Chapter XVIII
Utility–Cost Tradeoffs in the Design of Data Resources

Adir Even
Ben Gurion University of the Negev, Israel

G. Shankaranarayanan
Boston University School of Management, USA

Paul D. Berger
Bentley College, USA

ABSTRACT

This chapter introduces a novel perspective for designing and maintaining data resources. Data and the information systems that manage it, are critical organizational resources. Today the design and the maintenance of data management environments are driven primarily by technical and functional requirements. We suggest that economic considerations, such as the utility gained by the use of data resources and the costs involved in implementing and maintaining them, may significantly affect data management decisions. We propose an analytical framework for analyzing utility-cost tradeoffs and optimizing design. Its application is demonstrated for analyzing certain design decisions in a data warehouse environment. The analysis considers variability and inequality in the utility of data resources, and possible uncertainties with usage and implementation.

INTRODUCTION

Data, along with information systems and technologies (IS/IT) that manage it, is a critical organizational resource. Advances in data management technologies and the growing diversity of data sources allow firms to manage large data repositories and benefit from using them for enabling new business processes, supporting decision making, and generating revenue as a commodity. Different aspects of data management such as design, quality improvement, and integration into business processes have typically been studied from technical and functional perspectives. Economic aspects, such as the benefits gained from the use of data resources and the costs associated with managing them, have not been

explored in depth. In this chapter, we suggest that economic considerations significantly affect data management decisions, hence, deserve further examination. As investments in data resources grow, it is important to better understand their contribution to economic performance and business benefits. By conceptualizing business benefits as utility, we propose a framework for assessing and maximizing the contribution of data resources to the firm's economic performance.

We specifically link economic contribution to the design of data resources. Designs that improve capacity have higher utility contribution, but are often more expensive to implement. Enhancing design may require higher investments in IT and labor, increasing costs to the point of offsetting the utility gained. To what extent can design affect economic performance? Can maximizing performance direct design? These questions highlight a gap in data management research—while functional and technical aspects of design are well addressed, economic aspects are rarely explored. In this study, we identify design characteristics that impact utility-cost tradeoffs, model their economic effects, and use the models to assess design alternatives. We refer to this approach as *economics-driven design*. Importantly, we view administration and maintenance of information systems and data resources as an integral part of the implementation lifecycle. Hence, we use the term *design* to refer not only to the efforts involved in implementing entirely new systems or data resources, but also to the formulation of data and system administration policies and the implementation of data maintenance, improvement, and enhancement solutions.

We examine economics-driven design in the context of a data warehouse (DW)—an IS environment that manages large data archives. The high implementation and maintenance costs associated with a DW have been examined, but their business-value contribution has been rarely assessed. This study contributes by examining the utility of data resources in a DW. It introduces the concept of "utility inequality"—the extent to which items within a data collection differ in their business contribution. Understanding inequality and the associated utility-cost tradeoffs can help improve data management decisions from an economic standpoint. We link these tradeoffs to DW design decisions. Understanding economic effects of these decisions can improve design outcomes and help justify associated investments.

In the rest of this chapter, we first review the relevant background. We then lay the theoretical foundations of our framework for economic assessment of design alternatives, focusing on design decisions in a DW. We introduce the concept of utility inequality and develop quantitative tools for assessing it in large datasets. Utility inequality is shown to introduce economic tradeoffs. We analyze these tradeoffs and their implications for design decisions in data management environments. Acknowledging uncertainties with the utility gained, we then frame certain high-level design strategies as real-options investments. We further model economic effects of some design decisions along these strategies, describing conditions under which a certain strategy can turn out to be superior. We conclude by highlighting contributions and limitations of this study and suggest directions for further research.

RELEVANT BACKGROUND

Design is defined as teleological and goal-driven activity, aimed at the creation of new artifacts (Simon, 1996). Design research seeks to extend the boundaries of human and organizational capabilities by creating new and innovative artifacts (March & Smith, 1995). It is particularly important to field of Information Systems (IS) management, as the success and the impact of information systems significantly depend on their design (Hevner, March, Park, & Ram, 2004). March and Smith (1995) differentiate between behavioral research as "knowledge producing" and design research as "knowledge using." While the key contribution

of the former category is producing theories that explain behavior, the latter applies these theories in contexts of usage. This study offers new insights into the design of information systems, data management environments in particular, by linking design and maintenance decisions to economic tradeoffs.

Data offers benefits when used to support business operations and managerial decision making, and as a source for generating revenue. To use data effectively, organizations invest in repositories that store data resource, systems that process them, and tools that deliver data products to consumers. Today, the design and administration of data management environments are driven primarily by technical (e.g., storage space, processing speed, and monitoring capabilities) and functional requirements (e.g., the data contents, presentation format, delivery configuration). The need to address both these requirements affects the selection of information technologies, the design of data resources and the systems that manage them, as well as data management policies and procedures. Unlike technical and the functional perspectives, economic outcomes are rarely explored and addressed in the context of data management. As the volumes of data managed by organizations rapidly increase along with associated costs, we argue that understanding the economic impact of data management decisions is becoming no less important. To better understand the economic perspective of data management, we link the usage of data resources to utility and associate the implementation of systems and resources for supporting usage with costs. We suggest that certain data management decisions may influence both utility and cost and, hence, introduce significant economic tradeoffs. Importantly, we do not suggest that the economic view of design replaces the need to address technical and functional needs. However, assessing economic outcomes and tradeoffs in the design process can offer important insights for addressing these needs in the most cost-effective manner.

Research has rarely offered an explicit link between data management decisions and their economic outcomes. Design methodologies such as the entity-relationships modeling and the relational database design (Elmasri & Navathe, 2006) link functional requirement to technical design, but do not offer insights into their economic implications. Functional and technical link exists, to some extent, in data quality management (DQM) research. Data quality, at a high level, is defined as fitness for use (Redman, 1996). DQM studies offer different methodologies for improving quality such as error detection and correction and analysis of quality measurements (Redman, 1996; Pipino, Yang, & Wang, 2002). As data management environments involve multiple processing stages (e.g., data acquisition, cleansing, transformation, storage, and delivery), DQM studies often view them as data manufacturing processes that create information products (Wang, 1998). We adopt this process/product view for conceptualizing data management from an economic perspective. Our framework is influenced by a study by Ballou, Wang, Pazer, and Tayi (1998), which associates data quality targets and economic outcomes.

Information products contribute value through usage and experience (Shapiro & Varian, 1999). This value reflects benefits such as improvements in decision outcomes or willingness to pay (Ahituv, 1980). The value depends on the context of usage and requires successful integration with complementary organizational resources (Sambamurthy, Bharadwaj, & Grover, 2003), and is often subject to risk and uncertainty (Benaroch, Lichtenstein, & Robinson, 2006). The value contribution of information resources may be affected by their technical characteristics and/or by the design of the environment that manage them (Even, Shankaranarayanan, & Berger, 2007). The utility function maps the configuration of IS/IT attributes to tangible value within some specific usage (Ahituv, 1980). Utility mappings have been used for optimal configuration of quality attributes in complex data processes (Ballou et al., 1998), and for identifying an optimal design of datasets (Even, Shankaranarayanan, & Berger, 2007). In this study, we

examine the extent to which utility distribution in large datasets impacts data management decisions. We specifically examine the magnitude of utility inequality—whether utility contribution is the same for each record in a dataset, or concentrated in a relatively small number of records. To support this, we adapt statistical tools that are commonly used in economic and social welfare studies for analyzing inequality in large populations.

Data management environments (and information systems in general) involve a diverse set of cost components (West, 1994). We differentiate between two high level cost categories—fixed versus variable. The former represents costs that do not vary directly with data volume and the latter represents costs that monotonically increase with the volume. We view the management of data as involving four high-level steps: acquisition, processing, storage, and delivery. Costs, both fixed and variable, can be attributed to these activities. Some costs are common to all activities. For example, fixed cost associated with investments in system infrastructure (e.g., hardware, operating systems, and networking), software design and programming efforts, and managerial overhead, or variable costs associated with on-going monitoring and troubleshooting. Other costs are more specific to certain activities. For example, fees paid to a data vendor (West, 2000) can be interpreted as variable acquisition costs. Investments in ETL (extraction, transformation, and loading) tools and business-intelligence platforms in a data warehouse environment can be viewed as fixed processing cost and fixed delivery cost, respectively. Purchasing database management software is a fixed storage cost, while investing in storage capacity (i.e., disk space and hardware for managing it) can be viewed as a variable storage cost. We assume, in general, that the variable costs are linearly proportional to the data volume.

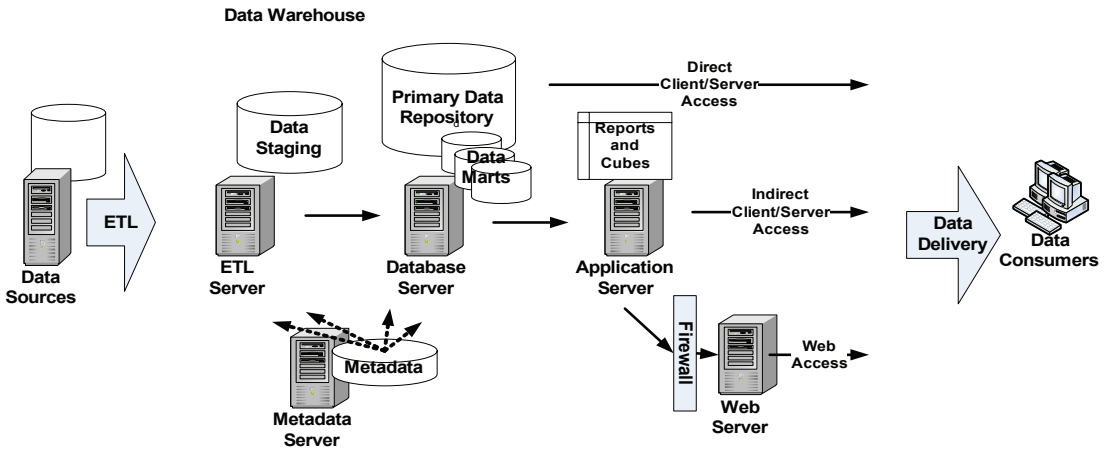
We view the goal of data management as maximization of economic performance. The criterion evaluated is the net-benefit—the difference between the overall utility and the overall cost. Increasing data volume and system capacity improves utility, but also involves higher costs. In certain cases, the marginal cost may more than offset the marginal utility; hence, maximizing volume and capacity may result in a sub-optimal net-benefit. We apply microeconomic modeling techniques to analyze these tradeoffs. Such models are commonly used for analyzing utility-cost tradeoffs and optimizing the benefit of products, processes, and services. Microeconomic models have been applied in IS/IT studies to a limited extent. Ballou et al. (1998) use utility-cost mapping for optimizing data quality targets and Even et al. (2007) apply microeconomic model for optimizing large datasets in a data warehouse. In this study, we develop microeconomic models that map the effect of increasing data volume and system capacity on utility and cost and use these models for evaluating tradeoffs and optimizing design decisions.

THE DESIGN OF A DATA WAREHOUSE

A data warehouse (DW) is an IS environment for managing large data repositories, aimed to support data analysis and decision making (Kimball, Reeves, Ross, & Thornthwaite, 2000). The implementation of these large complex environments involves major technical efforts and substantial managerial and organizational challenges (Wixom & Watson, 2001). Figure 1 describes the DW architecture at a high level. The key components of this architecture and the associated design challenges are further described in the following paragraphs.

- **Data acquisition:** A DW typically imports and integrates data from multiple sources. These sources are commonly other operational systems in the organization, but in certain cases they may by external (e.g., importing financial quotes from a market-data vendor).
- **Data staging:** The purpose of data staging is to prepare the data that was retrieved from a source for permanent storage. Data staging

Figure 1. A high-level data warehouse architecture



involves cleansing (i.e., detection and correction of quality defects), integration of data from multiple datasets, and transformation into new formats.

- **Primary data repository:** After staging, the data is loaded for permanent storage in the primary data repository. Datasets in a DW are often organized in star schemes. The "center of the star" is a fact table that contains measures of historical business transactions (e.g., price, quantity, and revenue). It is linked to multiple dimension tables, each representing a business dimension (or entity) that describes the transaction (e.g., client, product, location, and date). Fact tables are the largest datasets in the DW repository, and the number of records in such a table significantly affects the overall size of the repository. In this study, we demonstrate the economics-driven design principles for configuring of the number of records in a large dataset, such as a fact table.
- **Data marts:** A data mart is a dataset (or a collection of associated datasets), which contains data subsets from the main repository, restructured to address certain data usage needs. These data subsets are retrieved from the main repository and transformed into structures the permit efficient support for decision making and data analysis. The data marts are occasionally refreshed with new data and enhanced to support new usage requirements.
- **Data delivery and consumption:** A DW needs to provide utilities for retrieving data from the data repository and the data marts, transforming data to interpretable presentation formats (e.g., tables, charts), and delivering it to consumers (end-users, or other systems). Data usage typically involves specification of content (the underlying data), presentation, and delivery configuration (e.g., recipients and schedule). It also involves management of user authentication, as some users are not authorized to view certain data elements or to use certain reports and analysis tools. In today's DW environments, data delivery is typically managed with software platforms that permit rapid development of data products and efficient management of delivery schedules and user authorizations. Such platforms, commonly known as business intelligence (B.I.) tools, offer a variety of presentation capabilities, such as tables, charts, summary statistics, and advanced analytics. Some also permit interactive use, commonly known as OLAP (online analytical processing). Data de-

livery platforms often support multiple access modes—two-tier client-server configuration which permits access to data within the same network, and three-tier configuration that permits remote access. In certain cases, power-users are also authorized to access the data resources directly (e.g., via SQL queries).

- **Data maintenance and enhancement:** Due to the high volume and complexity, data in a DW is vulnerable to quality defects such as missing records and attribute values, inaccuracies due to data human errors or calculation mistakes, or changes in real-life entities that have not been captured. To support effective use and the ability to gain business benefits, data in the DW has to be constantly maintained and improved (Kimball et al., 2000). Further, the DW has to be frequently enhanced with new data entities and attributes to support new types of usage. Data enhancement and maintenance often requires investment in utilities for data monitoring, error detection, and automated correction. It also requires the development of maintenance procedures and policies.
- **ETL (extraction transformation loading):** A DW involves substantial data processing, including data acquisition, data transfer between stages, transformation to different structures and formats, and loading data into designated targets. A common term for data processing in a DW is ETL—extraction, transformation, and loading. ETL is commonly managed by dedicated software packages that permit rapid development of new processes (Kimball et al., 2000). ETL platforms also manage the ongoing execution of processes—scheduling, maintaining a certain sequence, monitoring, and evaluating the results. Many ETL platforms also provide utilities for data cleansing and automated error detection and correction.
- **Metadata:** Metadata is a high-level abstraction of data. It is essential for managing the different functionalities of DW subsystems such as data dictionary, documentation of database structure, process and delivery configuration, and user administration (Shankaranarayanan & Even, 2004). DW subsystems (e.g., database, ETL engines, and data delivery tools) may have different metadata requirements. The software platforms on which these subsystems are implemented typically provide some metadata management capabilities. Metadata requirements of different components are often inter-related—for example, mapping an ETL process, or configuring a data product, require knowledge of tables and attribute structure in the data resource layer. DW designers may therefore choose to enforce consistency between the different subsystems by centralizing metadata management and implementing a metadata repository that serves all applications.
- **Server configuration:** A DW typically requires a few servers, each addressing a different role: (a) *ETL server*: hosts the ETL engines and manages and executes back-end processing of data, (b) *database server*: manages the data resources, including the primary data repository and the derived data marts. This server requires installation of database management system (DBMS) software and sufficiently large storage space, (c) *application server*: manages utilities and tools for data retrieval, analysis and delivery, (d) *Web servers*: manages remote access via the Web, and (e) *metadata server:* manages the DW metadata layer. Each server should have sufficient storage and processing capacity to perform the role. A scalable design of these capacities is encouraged—the volumes of data that are managed in a DW are likely to grow over time, and with them processing and storage capacity needs. In DW environments that manage large-scale data resources, each of these roles is typically implemented with a different server (often, multiple servers per role). In relatively small environments,

different roles can be possibly addressed initially by the same server (e.g., one server managing ETL, metadata, and data-delivery applications), and as the DW grows the roles can be managed by specialized servers.

- **Security and access management:** Managing security and authorizations in a DW is critical. User groups are allowed to view only certain subsets of the data, and most users should not be allowed to access the system infrastructure, and/or to make changes to data resources and the derived data products. Managing the authorizations typically requires differentiation between three high-level roles: (a) developers—people who develop new components of the DW such as database schemas, ETL processes, reports, and other data products, (b) administrators—people who are in charge of the on-going operation of the DW, and (c) end users—people who are authorized to use the data product outcomes, but to make changes to the system, or interfere with its ongoing operation. Utilities for managing security and user authorizations are often offered by software development platforms (e.g., DBMS, ETL engines, and business-intelligence tools). Such utilities permit definitions of new users and user groups, assigning roles to different users and groups, and defining the authorizations for each role. Managing this authorization schema typically requires substantial metadata support. Firms often choose to centralize user authorization (e.g., the "single sign-on") for the DW and other organizational applications. This is accomplished by managing the associated metadata in a central repository and developing authentication utilities that are used consistently by all applications.

A key factor that derives and affects design decisions in a DW is the volume of the data resources managed. Rich and diverse data resources enhance usability and the utility potential. On the other hand, providing large data resources increases costs, as it requires investments in powerful servers, advanced software platforms (e.g., DBMS, ETL engines, and B.I. tools), and extensive maintenance and administration efforts. Though apparent, these tradeoffs between utility and cost are insufficiently addressed by today's DW design and management approaches. Introducing an economic perspective into existing DW methodologies, and developing analytical tools to aid with economic assessment, require explicitly and systematically linking data volumes and the associated design decisions to economic outcomes. For developing a more robust economic perspective for DW design, we next introduce a methodology for modeling and measuring the inequality of the utility in large data resources.

INEQUALITY IN THE UTILITY OF DATA RESOURCES

The utility of data reflects its business value contribution. Though all records have a similar structure, they may significantly vary in their content. We assume that this variability differentiates their relative importance to data consumers and, hence, their utility contribution in certain usage contexts. Given this variability, the overall utility in certain cases depends on the entire dataset, while in other cases it is affected only by a small subset of records. We interpret this as the magnitude of *utility inequality* in datasets. In this section, we lay the foundations for the utility inequality and develop analytical tools for modeling and assessing it in large datasets. In real-life, a dataset may have multiple usages, each with a different utility allocation. For brevity, we restrict our description here to a single utility allocation. The models and the analytical methods described can be extended to address multiple-usage scenarios as well.

We consider a tabular dataset with N records maximum utility of $u^D \geq 0$. Utility reaches maximum when the entire dataset is available and may reduce to some extent if some records are missing or unfit for use. We allocate the dataset utility u^D

among the records ($u_n \geq 0$ for record *[n]*), based on their relative importance for the evaluated usage. We assume utility-additivity with no interaction effects, hence, $u^D=\Sigma_n u_n$. A simple utility allocation may assign an identical value per record (i.e., a constant $u_n=u^D/N$). However, while easy to compute, this "naïve" allocation rarely reflects real life data consumption, as dataset records often significantly differ in their importance, and hence, in their utility contribution.

For a large dataset (large N), we represent the distribution of utility among the records as a random variable u with a known probability density function (PDF) *f(u)*. From the PDF we can calculate the mean $\mu=E[u]$, the cumulative distribution function (CDF) *F(u)*, and the percent point function (PPF, the inverse of the CDF) *G(p)*. In this section, we demonstrate the computations for the continuous *Pareto distribution*, commonly used in economic, demographic, and ecological studies. However, similar computations can be applied to many other statistical distributions, continuous or discrete (e.g., uniform, exponential, or Weibull). The Pareto distribution (Figure 2) is characterized by two parameters—the highest probability is assigned to the lowest possible value of $Z>0$ (Z can be arbitrarily close to 0). The probability declines as the value grows and the parameter $w \geq 1$ defines the rate of decline:

$$f(u)=wu^{-(w+1)}/Z^{-w} \quad u \geq Z, \quad F(u)=1-(u/Z)^{-w} \quad u \geq Z$$
$$G(p)=Z/(1-p)^{1/w}, \quad \mu = wZ/(w-1) \qquad (1)$$

To assess the extent to which dataset records differ in their utility contribution, we define R, the proportion of highest-utility records, as a *[0,1]* ratio between the N^* records of highest utility and N, the total number of records in the dataset (e.g., $R=0.2$ for a dataset with $N=1{,}000{,}000$ records is the ratio for the $N^*=200{,}000$ records that offer the highest utility). The *cumulative utility curve L(R)* is defined as a *[0,1]* proportion of the overall utility as a function of R. *L(R)* can be calculated from the percent point function *G(p)*. For a large N, the added utility for a small probability interval *[p, p+Δp]* can be approximated by $NG(p)\Delta p$ (Figure 2a). Taking $\Delta p \rightarrow 0$, integrating the PPF over *[1-R, 1]* (Figure 2b), and dividing the result by the total utility (approximated by μN), we obtain the cumulative utility curve *L(R)* (Figure 2c):

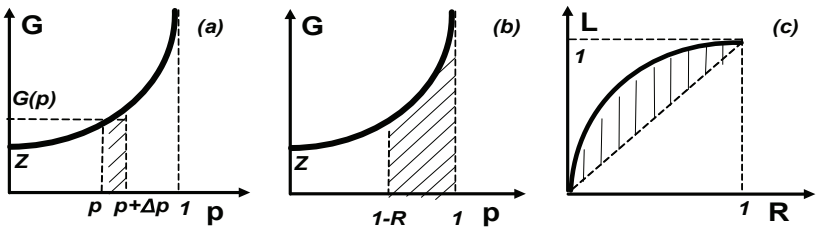
$$L(R)=\frac{N\int_{1-R}^{1} G(p)dp}{N\mu}=\frac{1}{\mu}\int_{1-R}^{1} G(p)dp \text{, where} \qquad (2)$$

- R – the *[0,1]* proportion of highest-utility records
- *L(R)* – the cumulative utility curve of the utility variable *u,* within *[0,1]*
- N – the number of dataset records
- *U,* μ – the utility variable and its mean
- *G(p)* – the percent point function of the utility variable *u*

As shown in *(3)*, *L(R)* does not depend on N or on the unit used to measure utility, but does depend on the utility distribution. The curve is defined for *[0,1]*, where *L(0)=0* and *L(1)=1*. The percent point function *G(p)* is monotonically increasing, hence, *L(R)* which is calculated by "backwards integration" over *G(p)* is monotonically increasing and concave within the *[0,1]* range. The first derivative of *L(R)* is therefore positive and monotonically decreasing, and the second derivative is negative. The maximum point of the curve (i.e., *L(1)=1*) represents the maximum possible dataset utility u^D, and the curve reflects the maximum portion of overall utility that can be obtained by the partial dataset—that is, when only a portion R of the dataset is available, the utility of $u^D L(R)$ can be achieved at best.

The cumulative utility *L(R)* is equivalent to the Lorentz curve, a statistical tool for modeling inequality in value distributions. Gini's coefficient, derived from the Lorentz curve, is commonly used to measure inequality. This coefficient *(φ)* measures

Figure 2. Obtaining the cumulative utility curve from the PPF by integration



the relative area between the curve and the *45°* line (i.e., *f(R)=R*). This area is highlighted in Figure 2c, and can be calculated by:

$$\varphi = \frac{\int_0^1 L(r)dr - \int_0^1 rdr}{\int_0^1 rdr} = 2\int_0^1 L(r)dr - 1 =$$

$$2\int_0^1 \left(\frac{1}{\mu}\int_{1-R}^1 G(p)dp \right) dr - 1 = \frac{2}{\mu}\int_0^1 pG(p)dp - 1 \qquad (3)$$

The value of φ is within *[0,1]*, where a higher value indicates a greater utility inequality. The lower bound, $\varphi \rightarrow 0$, indicates perfect equality—dataset records with identical utility and a curve that approaches *L(R)=R*. The upper bound, $\varphi \rightarrow 1$, indicates a high degree of inequality—a small portion of records with a relatively high utility, while the utility of most other records is substantially lower. The corresponding curve in this case approaches *L(R)=1*. The curve and the coefficient can be further evaluated for specific distributions and can often be expressed using a closed analytical form. For the Pareto distribution, the curve and coefficient are:

$$L(R) = \frac{1}{\mu}\int_{1-R}^1 G(p)dp = \frac{w-1}{wZ}\int_{1-R}^1 \frac{Z}{(1-p)^{1/w}}dp = R^{1-\frac{1}{w}}$$

$$\varphi = \frac{2}{\mu}\int_0^1 pG(p)dp - 1 = \frac{1}{2w-1} \qquad (4)$$

With the Pareto distribution, the curve and coefficient do not depend on the minimum value parameter *Z*, but only on the decline rate parameter *w*. Inequality decreases with *w*, where *w=1* indicates the highest possible inequality (*L(R)=1*, $\varphi=1$). Conversely, with $w \rightarrow \infty$, $L(R) \rightarrow R$, and $\varphi \rightarrow 0$. The utility in this case is nearly identical for all records, that is, *u=~Z* with probability of *~1*.

IMPLICATIONS FOR THE DESIGN AND THE MAINTENANCE OF DATA RESOURCES

Utility inequality among records may have implications for data management decisions. This effect can be evaluated from an economic perspective by assessing the effects of inequality on utility-cost tradeoffs and the overall net-benefit. We first consider utility-cost tradeoffs and net-benefit optimization for a single system configuration, assuming a linear cost model. We then extend the analysis to include decision scenarios that evaluate multiple system configurations.

For a single configuration, we consider *u*, the aggregated utility variable with corresponding maximum utility u^D, cumulative utility curve *L(R)*, and inequality coefficient φ. We define *U(R)*, the dataset utility curve (Figure 3a), as a function of *R*, the proportion of highest-utility records:

$$U(R) = u^D L(R), \text{ where} \qquad (5)$$

- *R* – the *[0,1]* proportion of highest-utility records
- *U(R)* – the maximal possible utility for *R*

- u^D – the maximal possible utility for the entire dataset (i.e., for $R=1$)
- $L(R)$ – the cumulative utility of the utility variable u, the aggregated utility variable

The cumulative utility curve $L(R)$, hence also $U(R)$, are monotonically increasing with a declining marginal return—a critical property that supports our argument of utility-cost tradeoffs. This property is explained by our definition of the proportion variable R as reflecting the sorting of records in a descending order of utility contribution.

Managing the dataset under a certain system configuration involves costs. We assume identical variable cost per record, uncorrelated to the record's utility—records are often acquired using the same acquisition tools, purchased at the same price, processed and monitored by the same back-end applications, and stored in the same format. Accordingly, we model the cost (for a given system configuration) as a linear curve (Figure 3b) with a variable component c^v, linearly-proportional to the dataset size (hence, to R), and a fixed component c^f, independent of the dataset size (we relax this assumption later, when evaluating multiple configurations):

$$C(R) = c^f + c^v R \text{ , where} \qquad (6)$$

- R – the *[0, 1]* proportion of highest-utility records
- $C(R)$ – the dataset cost for R
- c^f – a positive fixed cost
- c^v – a positive unit variable cost

Assuming that utility and cost are scaled to the same units, the net-benefit contribution $B(R)$ of the dataset is the difference between utility and cost (Figure 3c):

$$B(R) = U(R) - C(R) = u^D L(R) - \left(c^f + c^v R\right) \qquad (7)$$

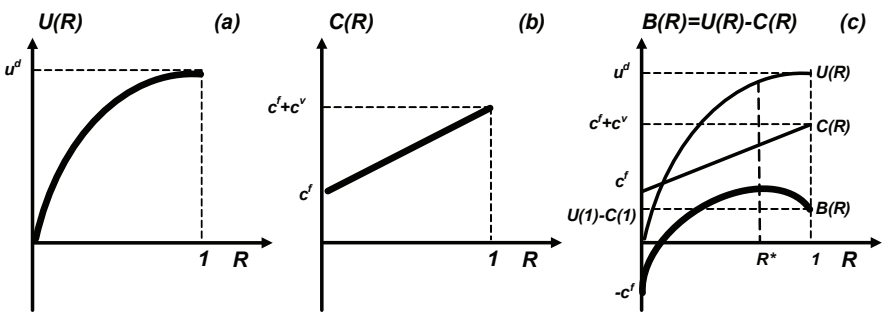
Due to c^f, $B(R)$ is negative at $R=0$ (the entire curve may be negative if the fixed cost is higher than the utility for all values of R). It is concave, and hence, has a single maximum within *[0, 1]*. An optimum R^{OPT} can be obtained by comparing the first derivative of *(7)* to *0*:

$$dB(R)/dR = u^D \left(dL(R)/dR\right) - c^v = 0, \text{ or}$$

$$dL(R)/dR = c^v/u^D \qquad (8)$$

Below R^{OPT}, the net-benefit can be improved by increasing R, since the added utility is higher than the added cost. Beyond R^{OPT}, the marginal cost exceeds the marginal utility and increasing R reduces the net-benefit. For a steep curve (i.e., a high degree of inequality, $L(R) \rightarrow 1$, $\varphi \rightarrow 1$), or when the variable cost is significantly higher than the maximal utility

Figure 3. The (a) utility, (b) cost, and (c) net-benefit curves



(i.e., $c^v >> u^D$), the optimum approaches a low record proportion (i.e., $R^{OPT} \rightarrow 0$). If no positive R^{OPT} exists, the dataset cannot provide a positive net-benefit due to the fixed cost c^f. Conversely, if the variable cost is relatively low (i.e., $c^v << u^D$), R^{OPT} is obtained at a high record proportion (i.e., $R^{OPT} \rightarrow 1$). When the cumulative curve approaches the 45^o line (i.e., $L(R) \rightarrow R$, $\varphi \rightarrow 0$), the solution will be at one of the edges—either at $R^{OPT}=0$, or at $R^{OPT}=1$. Whether the R^{OPT} solution is within the $[0,1]$ range or at the edges, a positive net benefit is not guaranteed and has to be verified.

The optimality equation *(8)* can be extended for specific utility distributions. For example, using the Pareto distribution's curve *(4)*, the optimum R^{OPT} can be obtained by:

$$dL(R)/dR = (1-1/w)R^{-1/w} - c^v/u^D, \text{ and}$$

$$R^{OPT} = \left[(1-1/w)u^D/c^v\right]^w \qquad (9)$$

For $w>1$, the optimum R^{OPT} for the Pareto distribution is always positive. It is within $[0,1]$ when the utility-cost ratio is $c^v/u^D \geq 1-(1/w)$, otherwise, the optimal net-benefit is obtained at $R^{OPT}=1$. The optimum approaches 0 for a high degree of inequality $(w \rightarrow 1)$, that is, when the majority of the utility is obtained from a relatively small fraction of records. The dependence of R^{OPT} on the utility-cost ratio grows with equality (i.e., greater w). When the variable cost is relatively very small (i.e., $c^v << u^D$), the optimal net-benefit is more likely to be obtained when the entire dataset is included ($R^{OPT}=1$). When the variable cost is substantially large, the optimal net-benefit is more likely to be at $R^{OPT}<1$. For a high a degree of inequality (i.e., $w \rightarrow \infty$), the expression $(1-1/w)^w$ converges to a constant $1/e$. If the utility is higher than the variable cost ($u^D > c^v$), $(u^D/c^v)^w \rightarrow \infty$, and the optimum is obtained for the entire dataset (i.e., $R^{OPT}=1$). If $u^D < c^v$, $(u^D/c^v)^w \rightarrow 0$, $R^{OPT} \rightarrow 0$, and the dataset is unlikely to yield any positive net-benefit.

Utility-cost trade-offs for Pareto-distributed utility can be also assessed at the record level. A Pareto distribution implies positive record utility: $Z>0$ (Eq. 1) represents the lowest-possible utility. If the variable-cost per record $c^{v,R}$ is always lower than Z, the added utility will always justify the added variable cost, hence, R will be maximized at 1 (it is still possible that the entire dataset will not be implemented if the fixed cost is too high). On the other hand, if $c^{v,R} > Z$, for low-utility records, the variable cost will offset the benefit and we are likely to get $R^{OPT}<1$.

Managerial Implications of Utility Inequality and Utility-Cost Tradeoffs

The presence of utility-cost tradeoffs, which may turn out to be significant in some cases, demands reexamination of key data management decisions. Key insights and recommendation are summarized in Table 1.

- **Dataset configuration:** When the utility distribution creates utility-cost tradeoffs, there is a need to carefully evaluate the economic effects of the design characteristics of the dataset (such as the targeted quality level, inclusion/exclusion of attributes, time span coverage and granularity) and design the dataset accordingly. For some datasets, the decision to implement can be an "all-or-nothing" decision (i.e., implement/manage the entire dataset, or avoid implementation altogether). Such datasets will be characterized by a low degree of inequality (i.e., $L(R)$ converging to the 45^o line and φ approaching 0). Datasets with high utility inequality (i.e., $L(R) \rightarrow 1$ and $\varphi \rightarrow 1$), are likely to benefit from differentiating records based on their relative utility contribution, and managing each subset differently. Depending on utility-cost tradeoffs, the designer may exclude low-utility records or manage them separately in a cheaper storage. A typical situation in real life data management is the archiving of older data. Less recent records are assumed to be

less important and, hence, administrators often exclude older records from actively used datasets. Modeling and assessing inequality by associating utility with the age of records can help configure the economically optimal time span in the dataset.

- **The design of data management environments:** The differential in utility values can possibly change the scope of data design, especially in environments that manage large datasets. The magnitude of inequality and the associated economic tradeoffs may affect the design of the entire data warehouse (DW). Large datasets require high investments in infrastructure (e.g., more powerful computers and database servers) and data delivery (e.g., data retrieval, reporting and business intelligence tools), as certain system configurations may have capacity limits on the volumes of data that they can effectively manage. Investment in powerful DW infrastructure will be harder to justify if the vast majority of the utility is obtained from a small fraction of the data, which can be effectively managed with a simple and inexpensive system configuration.
- **Data acquisition, retention, and pricing:** Utility inequality can impact data acquisition and retention policies. The value gained from purchasing and maintaining large data volumes is not always obvious. If data records can be differentiated based on utility contribution, it would make sense to acquire and maintain the records that contribute more to utility. Records with lower contribution may be archived or deleted, avoiding maintenance costs. Further, understanding the variability in utility can help define differential pricing policies by data vendors. Data vendors typically apply bulk pricing policies, based on technical characteristics such as data volume and the number of data retrieval activities required to serve the customer (West, 2000). If the utility distribution of data resources is better understood, vendors can develop more profitable pricing policies for data, based on its potential utility to the buyer.
- **Data quality management:** Utility differentiation can help define superior assessments of data quality along quality dimensions (e.g., completeness and accuracy). Such assessments reflect quality assessment in context (Even & Shankaranarayanan, 2007). Since the relative utility of a record can vary with usage contexts, purely objective measurements of data quality are not particularly beneficial. Data quality measurements that are based on relative utility consistently reflect contextual assessment, while complementing objective quality definitions. Further, differentiating the data records based on their utility contribution can help target quality monitoring and error detection efforts on the records that offer higher utility. This can help defining monitoring procedures and policies that are economically efficient.

EXTENDING THE SCOPE: REAL-OPTION DESIGN STRATEGIES

Our model for designing a tabular dataset assumed deterministic utility. In this section, we extend the framework to address design scenarios in which the utility gained from data usage is uncertain to some extent. To assess the affect of uncertainty on design decisions we apply real-options modeling. This modeling approach addresses the optimization of investment policies under conditions of uncertain outcome, where certain investments can be deferred to future stages. It has been applied in IS research for optimizing the investments in information technologies (Benaroch et al., 2006).

Two design aspects must be considered for addressing uncertainty:

- **Capacity:** To support usage requirements, an information system has to provide certain

resources and capabilities, which we refer to as its capacity. DW environments, for example, need to support different capacities in terms of data storage, processing, presentation, and delivery. These capacities are affected by design choices such as technology selection, data acquisition, and system configuration. Investing in slack capacity is a mechanism for addressing uncertainties. However, the cost of increasing capacity may be high, and when usage is relatively predictable the incentive to invest in slack capacity is low.

- **Timing:** A designer may consider deferring some investments to a later stage, until after some uncertainties are resolved. Postponing implementation can cause a delay penalty—implementing DW resources is often time-consuming and delaying the support of usage requirements might damage utility to some extent. Another drawback with deferring investments is a high switching cost. A certain system configuration can manage capacity effectively only to a certain limit. Increasing capacity beyond that limit may require switching to a new technology and discarding the old technology and writing off the associated initial investment. We link this interplay between capacity, deferred investments, delay penalties, and switching costs to design choices. Certain system configurations optimize the capacity to known needs with limited growth capability. Others allow some future growth with minimal delay penalties and switching costs, but involve a relatively high initial investment. We assume a two-stage design process. Some decisions and the associated investments are made at the initial stage. Others may be deferred, assuming that better decisions can be made after evaluating usage needs further.
- **Utility:** We represent utility as a binomial variable: *U* with a probability of *P*, or *0* with a probability of *1-P*. We consider two possible usage modes:
  a. **Exploitative:** Data consumption within relatively predictable business processes. Exploitation is assumed to contribute U^A utility with high certainty (i.e., probability of $P^A \approx 1$).
  b. **Explorative:** Business processes for which the utility contribution U^B has some uncertainty (i.e., $p^B < 1$). The designer may consider evaluating explorative usages further. After some evaluation time (T^E), and at some evaluation cost (C^E), the designer can assess whether

Table 1. The implications of utility-cost tradeoffs for data management

	High Inequality ($L(R) \rightarrow 1, \varphi \rightarrow 1$)	**High Equality ($L(R) \rightarrow R, \varphi \rightarrow 0$)**
Dataset Configuration	- Exclude or manage differently low-utility record.	- "All or nothing" decision—implement and manage the entire dataset, or avoid implementation altogether
DW Design	- Consider managing smaller, partial dataset, and reducing capacity requirements accordingly	- Invest in sufficient storage, processing, and delivery capacity to manage the entire dataset.
Data Acquisition, Retention, and Pricing	- Acquire and enrich only data subsets with high utility contribution - Consider removing lower-utility records, or archiving them separately - Consider differentiating pricing policies	- Apply similar acquire and enrichment policies to all data records - Allow equal access to the entire dataset - Charge equally for all data items
Data Quality Management	- Consider relative utility contribution when measuring data quality levels - Give higher priority to error detection and correction in higher-utility records	- Attribute equal weight to all records when measuring quality levels - Apply similar error correction and detection to the entire dataset

explorative usage will be successful (i.e., $U^B>0$), or not (i.e., $U^B=0$).

- **Delay penalty:** The utility might diminish to some extent with delay. We define the delay sensitivity *D(T)* as a *[0,1]* decreasing function that reflects the extent to which utility is affected by delay. With no delay, there is no utility reduction (i.e., $D(T=0)=1$). As the delay increases, the utility reduces and may diminish entirely if the time delay is too large (i.e., $D(T\rightarrow\infty)=0$). Different usages may have different sensitivities to delays. Accordingly, we assume two different delay sensitivities for exploitative and explorative usages (D^A and D^B respectively).
- **Technology xelection and cost:** Costs are attributed to investments in capacity. We consider three configuration alternatives, each providing a different capacity at a different cost:
  a. *high-end solution*—provides the highest capacity, sufficient to support all usages. It is costly (C^H) and takes a long time (T^H) to implement.
  b. *low-end solution*—provides a low capacity, sufficient to satisfy exploitative usages (but not explorative) at a low cost ($C^L<C^H$) and requires a short implementation time ($T^L<T^H$). It does not permit capacity extensions, and hence, supporting explorative usages mandates switching to an entirely new solution and losing the investment on the previous solution.
  c. *upgradeable solution*—permits implementing a relatively low capacity initially and increasing it later, if needed. We assume a time T^F for implementing the foundation for such a solution that has a cost of C^F. This foundation can support exploitation, but not exploration. The foundation is costlier ($C^L<C^F$) and takes longer to implement ($T^L<T^F$) than the low-end solution, but is cheaper ($C^F<C^H$) and less time-intensive ($T^F<T^H$) than the high-end solution. To support exploration the foundation needs upgrades, costing C^U and taking time T^U to implement. The combined cost and time of the upgradeable solution is higher than that for the high-end solution, ($C^F+C^U>C^H$, $T^F+T^U>T^H$). Hence, for a single-stage decision, the latter will be preferred.

Given these parameters, we evaluate four design strategies, show their utility-cost effects, and identify conditions under which a strategy can outperform the others. The first strategy, an investment in full-capacity made at the initial stage, is the baseline for comparison. The others, in which some investments are deferred, are viewed as real-options—the designer "buys the option" to decide later. The real-option net benefit (*RONB*) considers: *(1) real-option utility cifferential* (*ROUD*)—the gain (or loss) of utility due to the ability to deliver solutions earlier (or later), and *(2) real-option cost differential* (*ROCD*)—costs that are added or reduced by deferring the investment. A real-option strategy is preferred over the baseline if it increases the net benefit (i.e., $RONB=ROUD-ROCD>0$). We assume that the overall net benefit is positive for at least one strategy, otherwise the design initiative cannot be economically justified.

Maximize (M)

Here, the designer chooses to implement a high-end solution at the start with a cost of C^H. This solution permits gaining utility from both exploitation and exploration, but with some delay penalty due to the long implementation time (T^H). Applying (*1*), the anticipated net benefit B^M from this strategy is:

$$B^M = D^A\left(T^H\right)U^A + P^B D^B\left(T^H\right)U^B - C^H \qquad (10)$$

Switch (S)

With this strategy, the designer initially invests in a low-end solution that supports exploitative us-

age. After some evaluation time (T^E), the designer switches to a high-end solution if explorative usage can yield positive utility, or maintains the low-end solution otherwise. Applying (*1*), the anticipated net benefit B^S from this strategy is:

$$B^S = D^A\left(T^L\right)U^A + P^B D^B\left(T^E + T^H\right)U^B - \left(C^L + C^E + P^B C^H\right) \quad (11)$$

When compared with "maximize," some exploitative utility is gained due to a smaller delay-penalty ($T^L<T^H$), but some utility is lost due delaying exploration. The real-option utility differential is:

$$ROUD^S = \left[D^A\left(T^L\right) - D^A\left(T^H\right)\right]U^A - P^B\left[D^B\left(T^H\right) - D^B\left(T^E + T^H\right)\right]U^B$$

It is likely to be positive when the utility gained from exploitation is significantly higher that the utility gained from exploration ($U^A>>U^B$), when the probability of gaining utility from exploration is low ($P^B \rightarrow 0$), when the time saved by implementing a low-end solution is significant ($T^L<<T^H$), and/or when the evaluation time is relatively short ($T^E<<T^H$). The cost differential of this strategy is $ROCD^S = C^L + C^E - \left(1 - P^B\right)C^H$. It increases with the costs of a low-end solution (C^L) and evaluation (C^E), and decreases with a higher opportunity to save the cost of a high-end solution ($1\text{-}P^B)C^H$. This cost-saving opportunity will be significant when C^H is high, and/or when the probability P^B of gaining utility from exploration is low. However, there is a chance that the investment in a low-end solution will be wasted, and that the firm will need to reinvest in a high-end solution and suffer some delay penalty. This strategy will outperform "maximize" if its real-option net benefit ($RONB^S=ROUD^S\text{-}ROCD^S$) is positive:

$$RONB^S = \left[D^A\left(T^L\right) - D^A\left(T^H\right)\right]U^A - P^B\left[D^B\left(T^H\right) - D^B\left(T^E + T^H\right)\right]U^B - \left(C^L + C^E - \left(1 - P^B\right)C^H\right) \quad (12)$$

Upgrade (G)

Here, the designer initially invests in a foundation that allows gradual growth in capacity. After some evaluation of explorative usage (which consumes a time of T^E and costs C^E), the designer can decide whether or not to invest in capacity upgrades. Applying (*1*), the anticipated net benefit B^G from this strategy is:

$$B^G = D^A\left(T^F\right)U^A + P^B D^B\left(T^E + T^U\right)U^B - \left(C^F + C^E + P^B C^U\right) \quad (13)$$

Compared to "maximize," some exploitative utility is gained due to time savings and some explorative utility is lost due to delay. The corresponding real-option utility differential is:

$$ROUD^G = \left[D^A\left(T^F\right) - D^A\left(T^H\right)\right]U^A - P^B\left[D^B\left(T^H\right) - D^B\left(T^E + T^U\right)\right]U^B$$

This differential is likely to be positive when the utility gained from exploitation is significantly higher that the utility gained from exploration ($U^A>>U^B$), when the probability of gaining utility from exploration is low ($P^B \rightarrow 0$), when the time-saved by implementing the foundation solution is significant ($T^F<T^H$), and/or when the evaluation time is relatively short ($T^E<<T^H$). The cost differential of this strategy is $ROCD^G = C^F + C^E + P^B C^U - C^H$. It will increase with higher costs of the evolutionary solution (C^F, C^U) and the evaluation (C^E). It will decrease when a high-end solution is expensive, or when the probability of gaining utility from exploration is low. Overall, this strategy will outperform "maximize" if its real-option net-benefit $RONB^E$ is positive:

$$RONB^G = \left[D^A\left(T^F\right) - D^A\left(T^H\right)\right]U^A - P^B\left[D^B\left(T^H\right) - D^B\left(T^E + T^U\right)\right]U^B - \left(C^F + C^E + P^B C^U - C^H\right) \quad (14)$$

"Upgrade" has a similar decision structure as "switch;" however, it reduces risk to some extent. Here, the designer pays an initial "premium" by decreasing the exploitative utility ($T^F > T^L$, hence, higher delay penalty), and increasing the cost ($C^F > C^L$). The benefit, which materializes at the second stage, is a faster support for exploration ($T^U < T^H$) at a lower cost ($C^U < C^H$). Overall, the "upgrade" strategy will outperform "switch" when its net benefit is higher ($B^G > B^S$), that is, when the second-stage gain is higher than the premium paid at the first stage:

$$P^B\left[\left(D^B\left(T^E+T^U\right)-D^B\left(T^E+T^H\right)\right)U^B+\left(C^H-C^U\right)\right]>$$
$$\left[D^A\left(T^L\right)-D^A\left(T^F\right)\right]U^A+\left(C^F-C^L\right)$$

The gain (the left-hand side) is higher than the premium if explorative utility (U^B) and/or the chance of gaining it (P^B) are high. It is also higher if the time saved (T^U *vs.* T^H) is more significant than the time added for implementing the foundation (T^F *vs.* T^L), and/or if the cost differential between the high-end solution and the upgrade (C^H-C^L) is more significant than the cost differential between the foundation and a low-end solution (C^H-C^L).

Postpone (P)

In this case, the designer will avoid implementation at the first stage, but prefer to evaluate usage more extensively and then decide between a high-end solution and a low-end solution. Applying (*1*), the anticipated net benefit from this strategy B^P is:

$$B^p=\left(1-P^B\right)\left(D^A\left(T^E+T^L\right)U^A-C^L\right)+$$
$$P^B\left(D^A\left(T^E+T^H\right)U^A+D^B\left(T^E+T^H\right)U^B-C^H\right)-C^E \quad (15)$$

This strategy has some utility loss due to time delay. The real-option utility differential is:

$$ROUD^P=-\left[\begin{array}{l}\left(1-P^B\right)\left(D^A\left(T^H\right)-D^A\left(T^E+T^L\right)U^A\right)+\\ P^B\left(D^B\left(T^H\right)-D^B\left(T^E+T^H\right)\right)U^B\end{array}\right]$$

Depending on the evaluation time, this differential may turn out to be insignificant (if T^E is very short), or even positive (e.g., if $T^L+T^E<<T^H$). The cost differential is $ROCD^P=C^E-\left(1-P^B\right)\left(C^H-C^L\right)$. This strategy may significantly reduce the cost, compared to capacity maximization, if the evaluation cost (C^E) is relatively small, if the cost-margin between a high-end and a low-end solution (C^H-C^L) is high, and/or if the chance of gaining explorative utility is low (P^B). Under this strategy, there is no wasted investment besides the evaluation cost. This strategy will outperform "maximize" if its real-option net-benefit $RONB^P$ is positive:

$$RONB^P=-\left[\begin{array}{l}\left(1-P^B\right)\left(D^A\left(T^H\right)-D^A\left(T^E+T^L\right)U^A\right)+\\ P^B\left(D^B\left(T^H\right)-D^B\left(T^E+T^H\right)\right)U^B\end{array}\right]-$$
$$\left(C^E-\left(1-P^B\right)\left(C^H-C^L\right)\right) \quad (16)$$

When compared with "switch," "postpone" strategy offers the same performance with respect to explorative usage. However, for exploitation, it saves the wasted investment in a low-end solution, but at the expense of some delay penalty. It will outperform "switch" if the net benefit is higher ($B^P > B^S$), or

$$P^BC^L>\left[\begin{array}{l}D^A\left(T^L\right)-\left(1-P^B\right)D^A\left(T^E+T^L\right)-\\ P^BD^A\left(T^E+T^H\right)\end{array}\right]U^A$$

The cost saving (the left-hand side) may be more significant than the utility reduction (the right-hand side) if exploitative utility (U^A) is relatively low, and/or if the chance of gaining explorative utility (P^B) is high and the delay penalty is insignificant. "Postpone" will outperform "upgrade" if its net benefit is higher ($B^P > B^G$), that is, if the cost saving exceeds the utility loss:

$$C^F-\left(1-P^B\right)C^L-P^B\left(C^H-C^U\right)>$$
$$\left[D^A\left(T^F\right)-\left(1-P^B\right)D^A\left(T^E+T^L\right)-P^BD^A\left(T^E+T^H\right)\right]U^A+$$
$$P^B\left[D^B\left(T^E+T^E\right)-D^B\left(T^E+T^H\right)\right]U^B \quad (17)$$

Table 2 lists conditions under which a certain strategy outperforms others. The superiority is affected by exploitative versus explorative usages, decrease in utility due to time delays, and capacity and cost differentials between technologies. To demonstrate the use of our analysis framework, we now extend it to address a specific DW design scenario.

Design Scenario: Configuring the Database Server

Purchasing, setting up, and maintaining the DW infrastructure that provides the hardware, software, and network resources needed for establishing data repositories can be very expensive. Data storage is a key factor determining capacity, as a DW often manages very large datasets. Implementing storage capacity involves the purchase of database management systems (DBMS) server software (e.g., Oracle, MS-SQL), the hardware to support it (e.g., server, disk space), and labor. Here we model the effects of establishing storage capacity and the associated selection of DBMS, considering a single large tabular dataset (e.g., a fact table in a DW). Dataset size is a key factor (among others) that influences DBMS selection, here reflected by the *[0,1]* proportion of high-utility records (*R*). We assume that the utility for usage *[i]* follows a Pareto distribution with corresponding parameters (Z_i, w_i). For brevity, we denote the sensitivity factor $\alpha_i = 1\text{-}1/w_i$. The corresponding cumulative utility curve $L_i(R) = R^{\alpha_i}$ monotonically increases with *R*. Since $w_i \geq 1$, $0 \leq \alpha_i \leq 1$ and the marginal utility added, decreases with *R*. Usage *[i]* may require some minimum record proportion coverage φ_i, defining a set of lower-bound constraints: $\{\varphi_i \leq R \leq 1\}$. The extended utility model for usage *[i]* is:

$$U_i = P_i D_i(T) U_i^{Max} R^{\alpha_i}, \text{ where} \qquad (18)$$

- U_i, U_i^{Max} – utility contribution and its upper bound, respectively
- P_i, $D_i(T)$ – the probability of gaining utility, and the delay sensitivity, respectively
- R – the *[0,1]* proportion of highest-utility records, $\varphi_i \leq R \leq 1$
- α_i – utility-sensitivity parameter, $0 \leq \alpha_i \leq 1$
- φ_i – lower bound on record proportion

We consider *K* possible DBMS solutions (indexed by *k*), each requiring an implementation time of T_k and has a fixed cost of C^F_k (e.g., DBMS software and hardware). A larger *R* covered implies a larger number of records and a larger storage space. Assuming an approximately fixed number of records per time unit, the dataset volume will be linearly proportional to *S*. Thus, the variable cost for adding storage space increases linearly with the record proportion—$C^V_k R$, where C^V_l is the cost for covering $R=1$. We also assume an upper limit on storage capacity A_k for configuration *[k]* ($0 \leq R \leq, A_k$) as some DBMS are limited by the volume of data they can handle without performance degradation. Similar to the previous scenario, the designer may choose to evaluate usage further (evaluation time—T^E, and cost - C^E). Using *(1)*, the choice of DBMS configuration can be a design optimization—select a configuration (indexed by *k*) and record proportion *R* such that the net benefit is maximized:

$$B_k = U_k - C_k = \sum_{i=1}^{I} P_i D_i \left(T_k + ET^E\right) U_i^{Max} R^{\alpha_i} - C_k^F - C_k^V R - EC^E \qquad (19)$$

s.t. $0 \leq R \leq A_k$ per k, $\varphi_i \leq R \leq 1$ per i,

where

- B_k, U_k, C_k – overall net benefit, utility and cost, respectively
- T_k – implementation time
- C_k^F, C_k^V – fixed and variable implementation costs
- C^E, T^E – evaluation cost and time, respectively
- E – indicator of whether evaluation is performed (=*1*) or not (=*0*)
- A_k – data volume capacity *R*, U_i^{Max}, P_i, D_i, α_i, φ_i – same as in (18)

Table 2. Design strategies

	Maximize (M)	Switch (S)	Upgrade (G)	Postpone (P)
Investment Strategy	- Implement a high-end solution at the first stage	- Implement a low-end solution at the first stage - Switch to a high-end solution later, if required	- Implement the foundation for an upgradeable solution first - Upgrade later, if required	- Avoid implementation at first stage - Choose later between a low end and a high end solution
Superior to others when	- High utility can be gained from exploration - Small cost/time differential - Long and/or expensive evaluation	- Low level and/or low probability of explorative utility - Small delay penalty for postponing support for explorative usage - A low-end solution is significantly faster and cheaper than others	- Significant, but not high explorative utility - Foundation cost/time not significantly high compared to low-end - The overall cost/time (foundation + upgrade) margin from a high-end solution is insignificant	- Evaluation can be performed in a fast and cheap manner - The cost and time of implementing a low-end solution are significant - The probability of success with explorative usage is relatively high

This design scenario involves a discrete set of choices (DBMS selection) and a continuous variable *(R)*. Solving a mixed-integer optimization model requires repeating the optimization of the continuous component iteratively for all possible discrete configurations, and choosing the DBMS solution that yields the highest net-benefit, along with the corresponding optimal setup.

Illustrative example: A hospital evaluates a DW for analyzing treatment history (e.g., doctor visits, treatments, medications, and labs). Possible usages evaluated are: (a) exploitative—ongoing reporting and monitoring such as inventory tracking, resource utilization, and treatment history for specific patients. These usages require a minimum R of φ^A. Their utility (U^A) has sensitivity of α^A, and delay sensitivity of $D^A$and (b) explorative—advanced analyses such as detecting shifts in resource utilization, identifying patterns of reactions to drugs, and segmenting treatment history along demographic and socioeconomic attributes. These usages require a minimum R of φ^B ($>\varphi^A$). Their anticipated utility is U^B with a success probability $P_b<1$, sensitivity of α^B, and a delay sensitivity, D^B. Covering the entire dataset (R=1) implies a larger data volume and the designer considers two database configurations. A low-end solution can be implemented within a short time (T^L) at low fixed/variable costs ($C^{F/L}$ and $C^{V/L}$ respectively). The capacity limit of this solution (A^L) permits exploitative usage, but not explorative. A high-end configuration can be implemented within a time of T^H, and at fixed/variable costs of $C^{F/H}$ and $C^{V/H}$ respectively. Its capacity (A^H) can support all usage types.

Based on these parameters, which DBMS configuration must be chosen, and what is the optimal record proportion coverage for this choice? Using (19), we evaluate different possible configurations considering the high-level strategies:

Maximize (M): Implement a high-end solution with no further evaluation (i.e., E=0) and cover a record proportion large enough to support all usages. The corresponding net-benefit is:

$$B^M(R) = D^A\left(T^H\right)U^A R^{\alpha^A} + P^B D^B\left(T^H\right)U^B R^{\alpha^B} - \left(C^{F/H} + C^{V/H} R\right)$$
$$\text{s. t. } \varphi^B \le R \le 1 \qquad (20)$$

This strategy is preferred if advanced analysis can lead to high utility gains with a substantial level of certainty, the evaluation cost is high, and/or the

delay penalty is too severe. The advantage will also depend on the maximum net-benefit obtainable by optimizing the record proportion *R*. Candidates for R^{OPT} can be obtained by comparing the first derivative of (20) to *0*:

$$\alpha^A D^A \left(T^H\right) U^A R^{\alpha^A - 1} + \alpha^B P^B D^B \left(T^H\right) U^B R^{\alpha^B - 1} = C^{V/H} \quad (21)$$

The left-hand side of (21) represents the marginal utility as a function of *R*. Since α^A and α^B are within *[0,1]*, this margin is positive at $R = \varphi^B$ and decreases to $\alpha^A D^A \left(T^H\right) U^A + \alpha^B P^B D^B \left(T^H\right) U^B$ as *R* approaches *1*. The right-hand side of (21) is a positive constant with respect to *R*, representing a fixed marginal cost. As the left-hand side is monotonically decreasing, (21) has a single solution at most, within *[0,1]*. To validate the optimality of the candidate solution (if it exists and lies within the $[\varphi^B,1]$ range), the second derivative of (20) must be evaluated:

$$\partial^2 B^M(R)/\partial R^2 = \left(\alpha^A - 1\right)\alpha^A D^A \left(T^H\right) U^A R^{\alpha^A - 2} + \left(\alpha^B - 1\right)\alpha^B D^B \left(T^H\right) U^B R^{\alpha^B - 2} \quad (22)$$

Since α^A and α^B are within *[0,1]*, the second derivative is negative and a candidate solution is a local maximum. If no solution exists within the $[\varphi^B,1]$ range, two edge conditions are possible: (a) the marginal-utility exceeds the marginal-cost, hence, *R=1* and (b) the marginal-utility is always lower, hence, $R = \varphi^B$.

Switch (S): with this strategy, the designer first implements a low-end solution and optimizes the record proportion coverage for exploitative usage. After some evaluation, the designer may switch to a high-end solution and re-optimize *R* accordingly. The evaluation of net-benefit and the corresponding *R* has to consider two possible scenarios:

1. Keeping a low-end solution at the later stage, with a corresponding net-benefit of:

$$B^{S/1}(R) = D^A \left(T^L\right) U^A R^{\alpha^A} - \left(C^{F/L} + C^{V/L} R + C^E\right), \text{ s.t. } \varphi^A < R < A^L \quad (23)$$

   The record proportion coverage in this case ($R^{OPT/1}$) can be calculated by following the methodology described above under the "maximize" strategy.

2. *Switching to a high-end solution at the second stage,* assuming that $R^{OPT/1}$ will be covered at the first. The net-benefit depends on a different optimal *R* ($R^{OPT/2}$), obtained from:

$$B^{S/2}(R) = D^A \left(T^L\right) U^A R^{\alpha^A} + D^B \left(T^E + T^H\right) U^B R^{\alpha^B} - \left(C^{F/L} + C^{V/L} R^{Opt/1} + C^E + C^{F/H} + C^{V/H} R\right), \text{ s.t. } \varphi^B < R < 1 \quad (24)$$

The former scenario has an occurrence probability of $(1\text{-}P^B)$ and the latter has an occurrence probability of P^B. The expected net-benefit from this strategy is:

$$B^S = \left(1 - P^B\right) B^{S/1}\left(R^{Opt/1}\right) + P^B B^{S/2}\left(R^{Opt/2}\right) \quad (25)$$

This strategy is preferred if implementing a low-end DBMS solution is significantly faster and cheaper, and/or if the expected utility from advanced analysis is relatively low or has low probability of obtaining. This solution will also be advantageous when the penalty for delaying support for explorative usage is relatively small.

Upgrade (G): Here, the designer will invest in the infrastructure for a high-end DBMS, but will initially optimize S for exploitation. After evaluation, the designer may consider increasing S to support all usages. The evaluation must consider two possible scenarios:

1. Keeping R coverage at a low level ($R^{OPT/1}$) and supporting exploitative usage only:

$$B^{G/1}(R) = D^A \left(T^H\right) U^A R^{\alpha^A} - \left(C^{F/H} + C^{V/H} R + C^E\right), \text{ s.t. } \varphi^A < R < 1 \quad (26)$$

2. Enlarging R coverage at the second stage to support explorative usage as well. Unlike "switch," with this strategy the investments made at stage one are not wasted. The net-benefit and the corresponding optimal $R^{OPT/2}$ can be obtained from:

$$B^{G/2}(R) = D^A\left(T^E\right)U^A R^{\alpha^A} + P^B D^A\left(T^H\right)U^B R^{\alpha^B} - \left(C^{H/L} + C^{H/L}R + C^E\right)$$

, s.t. $\varphi^B < R < 1$ (27)

Considering the occurrence probabilities of both scenarios, the expected net-benefit is:

$$B^G = \left(1 - P^B\right)B^{G/1}\left(R^{Opt/1}\right) + P^B B^{G/2}\left(R^{Opt/2}\right) \quad (28)$$

This strategy is advantageous if implementing the foundation is fast and inexpensive, but the marginal cost of increasing R is high. This is likely, for example, if data has to be acquired at a high cost, or if improving its quality is expensive.

Postpone (P): with this strategy, the designer first evaluates usage needs and accordingly decides on the solution. This assessment must consider two cases:

1. Implementing a low-end solution and maintaining R at a low level ($R^{OPT/1}$):

$$B^{P/1}(R) = D^A\left(T^E + T^L\right)U^A R^{\alpha^A} - \left(C^{F/L} + C^{V/L}R + C^E\right)$$

, s.t. $\varphi^A < R < A^L$ (29)

2. Implementing a high-end solution and maintain R at a high level ($R^{OPT/2}$):

$$B^{P/2}(R) = D^A\left(T^E + T^L\right)U^A R^{\alpha^A} + D^A\left(T^E + T^H\right)R^{\alpha^B} - \left(C^{F/H} + C^{V/H}R + C^E\right)$$

, s.t. $\varphi^B < R < 1$ (30)

Considering the occurrence probabilities of both scenarios, the expected net-benefit is:

$$B^P = \left(1 - P^B\right)B^{P/1}\left(R^{Opt/1}\right) + P^B B^{P/2}\left(R^{Opt/2}\right) \quad (31)$$

This strategy is preferred if the evaluation is fast and inexpensive. It is also likely to be better if the implementation cost and time for a low-end solution are relatively high, and/or if the probability of the explorative usage being successful is significant.

Storage capacity is a key design decision in DW environments. A large capacity might turn out to be very expensive in certain scenarios—DBMS software that supports high volume storage is expensive to license and requires considerable labor to configure and maintain. Alternately, a cheaper DBMS may be sufficient for most usages, but might limit storage capacity and consequently the ability to enhance the DW to support new information products and usages. The record proportion covered within the storage capacity limit has to be carefully evaluated, as it can significantly impact utility and the costs of data acquisition and maintenance.

CONCLUSION

The design and the maintenance of data management environments are resource-intensive. As the volumes of data managed and the associated costs increase, we advocate the need to examine data management from an economic perspective. We do not minimize the importance of satisfying technical and functional requirements in the design process but argue that design decisions must be economically justified. The perspective proposed will effectively supplement the traditional design approaches. We believe that this study is a first step in incorporating economic considerations into design of data management environments.

This study contributes to the design of a data warehouse by developing a framework for evaluating design decisions within. We explore the link between design decisions and economic benefits, suggesting that modeling the effects of these deci-

sions on economic outcomes can enhance design processes. We link economic tradeoffs and the related data design and administration decisions to the utility distribution of datasets, and develop analytical tools for assessing utility inequality. Utility is attributed to data resource by understanding usage in different business contexts. The need to support both exploitative usages (with relatively certain outcome) and explorative (with uncertain outcomes) usages has important design implications for the data management environment that supports these. Each usage type differs significantly in its data utilization patterns, and the differences can direct design decisions. Similarly, design decisions are linked to costs, as higher capacity, faster performance, and sophisticated capabilities require larger investments. Modeling the effect of design decisions on utility and costs can help assess tradeoffs and identify economically optimal designs. We analyze these tradeoffs in the context of designing the warehouse capacities, treating certain strategies as real-option investments. Increasing DW capacity to certain layers allows timely and less expensive support for new usages, thereby enhancing utility. Our evaluation addresses common DW design tradeoffs—investing in expensive slack capacity for explorative usages that offer a larger potential for utility gains but, at a substantial risk, versus optimizing capacity for more certain exploitative usages. We demonstrate these tradeoffs for some DW design scenarios and identify conditions in which a certain design strategy will outperform others.

The framework proposed here offers several opportunities for further research. It develops inequality measurements specifically for the Pareto distribution. Other distributions (continuous or discrete) may better reflect real-life utility distributions and should be explored. The cost model assumes an equal variable cost per dataset records. However, depending on the business setting, variable costs may not be linear with number of records, and the cost model might have to be revised. The current model applies sum-additive modeling of utility and cost factors, suggesting that the factors are independent and orthogonal. Value enhancement or neutralizing relationships may exist among utilities and costs, in which case the whole is not necessarily an additive sum of the parts. The framework needs to be enhanced to model such interactions and dependencies. Estimation of utility may turn to be an even greater challenge in business settings. Data resources contribute value through usage when embedded within business processes together with complementary resources such as financial investments, physical assets, and human expertise. It is difficult to attribute a utility contribution to data in complex business settings, isolated from business processes and complementary resources, and to allocate the utility across dataset records. Further, a data resource can be used by multiple consumers, each with different, and possibly conflicting, utility assignments. Adding multiple usage contexts adds yet another dimension to the complexity. Further, some usages are unknown when the dataset is established. Considering all these difficulties, methods for estimating utility and allocating it across records certainly require a more in-depth examination in real life settings.

To conclude, we state that design for improved economic performance is important not only for data management but also for system design. Our evaluation framework and the factors examined (the expected outcome and associated uncertainty, the differences in cost and performance across alternate technologies, and the negative impact of time delays) can be applied to design scenarios in other IS/IT environments. The magnitudes of utility-cost tradeoffs associated with design decisions are often high and quantitative evaluation of design alternatives can help identify economically-optimal choices. This in turn, can guide investment decisions regarding technology solutions. Our framework, which emphasizes economic considerations as goals that direct design and impact architectural choices, assists by laying a better foundation of the economic perspective for IS design.

REFERENCES

Ahituv, N. (1980). A systematic approach towards assessing the value of information system. *MIS Quarterly, 4*(4), 61-75.

Ballou, D. P., Wang, R., Pazer, H., & Tayi, G. (1998). Modeling information manufacturing systems to determine information quality. *Management Science, 44*(4), 462-484.

Benaroch, M., Lichtenstein, Y., & Robinson, K. (2006). Real-options in information technology risk management: An empirical validation of risk-option relationships. *MIS Quarterly, 30*(4), 827-864.

Elmasri, R., & Navathe, S. B. (2006). *Fundamentals of database systems* (5th ed.). Redding, MA: Addison Wesley.

Even, A., & Shankaranarayanan, G. (2007). Assessing data quality: a value-driven approach. *The Data Base for Advances in Inf. Systems, 38*(2), 76-93.

Even, A., Shankaranarayanan, G., & Berger, P. D. (2007). Economics-driven data management: An application to the design of tabular datasets. *IEEE Transactions on Knowledge and Data Engineering, 19*(6), 818-831.

Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. *MIS Quarterly, 28*(1), 75-105.

Kimball, R., Reeves, L., Ross, M., & Thornthwaite, W. (2000). *The data warehouse lifecycle toolkit*. New York: Wiley Computer Publishing.

March, S. T., & Smith, G. F. (1995). Design and natural science research on information technology. *Decision Support Systems, 15*, 251-266.

Pipino, L. L., Yang, W. L., & Wang, R. Y. (2002). Data quality assessment. *Communications of the ACM, 45*(4), 211-218.

Redman, T. C. (1996). *Data quality for the information age*. Boston: Artech House.

Sambamurthy, V., Bharadwaj, A., & Grover, V. (2003). Shaping agility through digital options: Reconceptualizing the role of information technology in contemporary firms. *MIS Quarterly, 27*(2), 237-263.

Shankaranarayanan, G., & Even, A. (2004). Managing metadata in a data warehouse: Pitfalls and possibilities. *Communications of the Association of Information Systems (CAIS), 14*(13), 1-49.

Shapiro, C., & Varian H. R. (1999). *Information rules*. Cambridge, MA: Harvard Business School Press.

Simon, H. A. (1996). *The science of the artificial* (3rd ed.). Boston: The MIT Press.

Wang, R. Y. (1998). A product perspective on total quality management. *Communications of the ACM, 41*(2), 58-65.

West, L. A., Jr. (1994). Researching the cost of information systems. *Journal of Management Information Systems, 11*(2), 75-107.

West, L. A., Jr. (2000). Private markets for public goods: Pricing strategies of online database vendors. *Journal of Management Information Systems, 17*(1), 59-84.

Wixom, B. H., & Watson, H. J. (2001). An empirical investigation of the factors affecting data warehousing success. *MIS Quarterly, 25*(1), 17-41.

KEY TERMS

Business Intelligence (B.I.) Tools: Software tools for reporting and data analysis in DW environments (e.g., business objects, cognos, and microstrategy). B.I. tools typically provide utilities for retrieving data, transforming in into different presentation formats (e.g., charts, tables, reports), and applying different forms of analysis (e.g., aggregation, statistics, data mining) towards decision making.

Data Structure: Data repositories follow a typical hierarchical structure. The lowermost-level of this hierarchy is the data-item, or the datum, the atomic data entity. The data-item is defined as a triplet <a,e,v>. The data value 'v' is selected from the value-domain associated with attribute 'a' of entity 'e,' which represents a physical or conceptual real-world object. The data-record is a collection of data-items that represent a set of attributes of an entity instance. A dataset is a collection of records that belong to the same instance type (e.g., a subset of records in a table), and a database is a collection of datasets with meaningful relationships among them. A data repository typically manages multiple databases, each serving different business purposes.

Data Utility, Cost, and Net-Benefit: Economic-performance measures, commonly expressed in monetary units. Utility reflects of the business value attributed to data within specific usage contexts. Cost reflects investments made in establishing data resources and the systems that manage them—for example, in acquisition, processing, storage, and delivery. The net-benefit is the difference between the overall utility and the overall cost.

Data Warehouse (DW): An IS environment that manages large data resources. The DW offers a broad and integrated view of the firm along different business functionalities, towards better support for managerial decision making. This is achieved by integrating data from multiple data sources, accumulating it over a long period of time, and providing utilities for efficient retrieval, aggregation, presentation and delivery of large amounts of data.

Extraction, Transformation, and Loading (ETL): A common name for back-end application for processing data in a DW. An ETL application provides utility for extracting and integrating data from different data sources (e.g., text files, RDBMS, on-line data feeds), transforming it to a different format, and loading it into a target data repository.

Real Options: A methodology for analyzing irrecoverable investments when outcome is uncertain. The investor may decide to make a smaller investment at current stage (perceived as "buying an option"), which permits deferring the decision on a full-scale investment to a later stage, when some of the uncertainly is resolved.

Star Schema, Facts, and Dimensions: Tabular datasets in a DW database are often organized in a "star schema." The fact table (the "center of the star") contains measures of historical business transactions (e.g., price, quantity, revenue). It is linked to multiple dimension tables, each representing a business dimension (or entity) that describes the transaction (e.g., client, product, location, and date).

Tabular Datasets, Relational Dataset Modeling, and RDBMS: The table is a commonly-used two-dimensional data model, which represents data-attributes as columns (a.k.a. fields) and entity-instances as rows (a.k.a. records). The relational database model maps data that represents business entities and the relationships between them into a collection of tabular datasets. The relational model underlies RDBMS (relational database management system) technologies for database management (e.g., Oracle, MS-SQL, and Sybase). Tabular datasets are also used in other common data-storage technologies, such as flat-files, spreadsheets, and statistical packages.

Glossary of notations

X	A vector of design characteristics
B	Net Benefit
U, u	Utility
C, c^F, c^V	Cost, Fixed Cost, Variable Cost
I, $[i]$	Number of usages, Corresponding index
J, $[j]$	Number of cost factors, Corresponding index
N, $[n]$	Number of records, Corresponding index
$f()$	Probability density function (PDF)
$F()$	Cumulative distribution function (CDF)
$G()$	Percent-Point function (PPF)
M	The mean of a random variable
Z, w	The parameters of a Pareto distribution
R	The proportion of highest-utility records
$L(R)$	The cumulative utility curve
Φ	The utility inequality coefficient
K, $[k]$	Number of databases, Corresponding index
P	Probability
T	Time, Time-delay
$D(T)$	Delay-penalty, as a function of time
$ROUD$	Real-Option Utility Differential
$ROUC$	Real-Option Cost Differential
$RONB$	Real-Option Net Benefit
$()^D$	Dataset-level variables
$()^A$	Exploitative usage variables
$()^B$	Explorative usage variables
$()^E$	Evaluation variables
$()^L$	Low-end technology variables
$()^H$	High-end technology variables
$()^F$	Foundation of upgradeable technology variables
$()^U$	Upgrade of upgradeable technology variables
$()^M$	“Maximize” strategy variables
$()^S$	“Switch” strategy variables
$()^G$	“Upgrade” strategy variables
$()^P$	“Postpone” strategy variables