

REAL-D: A Schema for Data Warehouses

Daniel E. O’Leary
University of Southern California

ABSTRACT: This article integrates McCarthy’s REA (Resources-Events-Agents) model and the closely related REAL (Resources Events Agents Locations) model with general capabilities and requirements of data warehouses. REA/REAL contribute a theory for capturing information about events and a focus on control relationships. Data warehouses bring time-period information and a focus on information facilitating the creation of value. Using aspects from both camps, a hybrid schema is developed called REAL-D, REAL for Data warehouses. Existing data warehouse approaches lack theory, while REA/REAL are theory based. Unique demands on data warehouses, however, make additional requirements on REA/REAL, including (1) addition of *time period* as another dimension in order to allow rollups from hour to day to week to month to year, (2) addition of location to facilitate rollups from office to city to district, (3) change from a pure *location* dimension to a nonhomogeneous dimension that allows rollup from person to office, and (4) change of relationships of agents from one of control to a marketing-oriented one.

Key Words: REA, REAL, REAL-D, Databases, Data warehouses.

I. INTRODUCTION

This article argues that REA/REAL’s theoretical model of resources, events, agents and locations provides a basis for a data warehouse¹ schema but that the unique requirements of data warehouses must be accounted for in order to meet the demands placed on data warehouses. This article merges contributions from the REA/REAL schema and the data warehouse schema as a basis for generating a revised schema for data warehouses, referred to as REAL-D.

REA/REAL and Data Warehouses

Databases conforming to McCarthy’s (1979, 1982) REA model and Denna et al. (1993) and Hollander et al.’s (1996) REAL model are designed to capture and parsimoniously preserve event information for a particular time period, typically a year or two. REA/REAL provide a theory for the information that needs to be captured about economic events. Consequently, the REA/REAL design ensures that periodic decision-making information is *captured*. REA is designed to provide an online event processing (OLEP) capability for accounting and other data, expanding on online transaction processing (OLTP). OLEP systems, such as REA/REAL databases, are updated with every transaction

¹ URLs for data warehousing are listed in the appendix.

that occurs in an organization. OLEP systems make changes in their enterprise databases with each event. OLEP systems differ from OLTP systems by capturing events not normally captured in transaction processing systems (e.g., sales call events) and by including information not typically captured in transaction processing systems (e.g., agent and location information).

Since about 1995, specialized databases known as data warehouses have been developed to *deliver* data to users for *analysis* of trends over time periods, products, locations, and other dimensions with the capability of online analytic processing capability² (OLAP). OLAP software provides the ability to access analytical data, such as time series, so that users may analyze the data themselves. To support OLAP, data warehouses, integrated databases designed to support decision making, provide large amounts of data to decision makers. Because they allow decision makers to query the database about customer, vendor and supplier behavior over time periods, data warehouses have been used by companies to help them compete (Appleton 1996) and thus create value (e.g., Darling 1996).

REA provides a theoretical model for capturing event information, while data warehouses provide a model for delivery of the data to users. Capture and delivery of event information are related because only captured data can be delivered. OLTP/OLEP databases typically form the base of data warehouses, providing snapshots of an enterprise's events on a daily, weekly, monthly, or yearly basis (Edelstein 1995). With updates occurring as frequently as daily, there is little time to provide OLTP/OLEP with increased content information. The same type of information however, can be provided for multiple periods. Therefore, the design of the OLTP/OLEP database is critical to the information that is captured for OLAP databases.

REAL-D

Data warehouses, however, have data requirements that are beyond those of the event information captured by REA/REAL. Data warehouses store multiple snapshots of transaction and/or event databases over multiple time periods and require the ability to drill down or up. Further, data warehouses are frequently designed to deliver information useful for marketing and other value-creating opportunities. As a result, a data warehouse schema must allow (1) the addition of time period as a dimension in order to allow rollups from hour to day to week to month to year, (2) the change from a pure location dimension to a non-homogeneous dimension that allows rollup from person to office (known as cumulating nonhomogeneity), and (3) a change of relationships of agents from one of control to a marketing-oriented one.

As a result, this article proposes integrating aspects of REA/REAL and data warehouses in order to establish a theory-based approach for data warehouse schema, REAL for Data warehouses (REAL-D), depicted in figure 1.

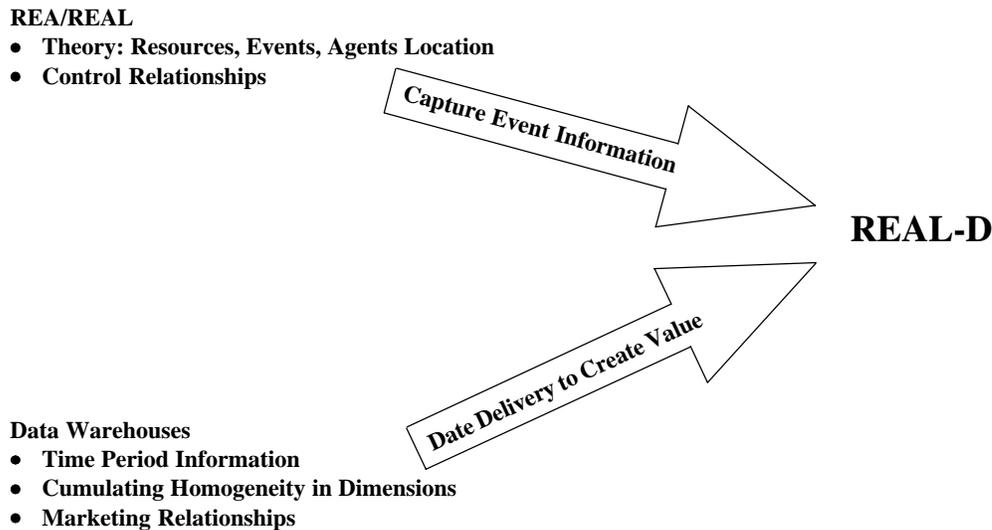
Motivation for New Data Warehouse Schema

The need for new schema for data warehouses is illustrated by a fatally flawed data warehouse project (Paul 1997), whose failure was attributed to difficulties in determining user requirements for a business model. Developers did not have any particular theory for generating data warehouse system requirements, users complained that interviews took too long, and potential users each had different views of the data, which began to obscure the business model. In the absence of a theoretical model to guide schema design, appropriate event information was not identified. Although REA/REAL provides a theoretical basis that ensures that event information is captured, REA/REAL do not address the following data warehousing issues:

- *Users.* In OLTP/OLEP a programmer typically generates a query that will be used thousands of times, whereas in data warehousing the user formulates and submits queries, possibly only once.
- *Content Needs.* Where OLTP/OLEP are designed for parsimony, data warehouses often contain aggregated information.
- *Data in Database.* OLTP and OLEP are typically designed to capture data for a firm for one or two years, but OLAP is designed to provide users with that same type of information over multiple

² URLs for OLAP are listed in the appendix.

FIGURE 1
Contributions of REA/REAL and Data Warehouses to REAL-D



periods, such as three to five or more years.

- *System.* Although data warehouse users might be able to access usable data through views, it is highly likely that a user's ill-framed queries will grind the database to a halt if the views are derived from a transaction-oriented database (SAS date).

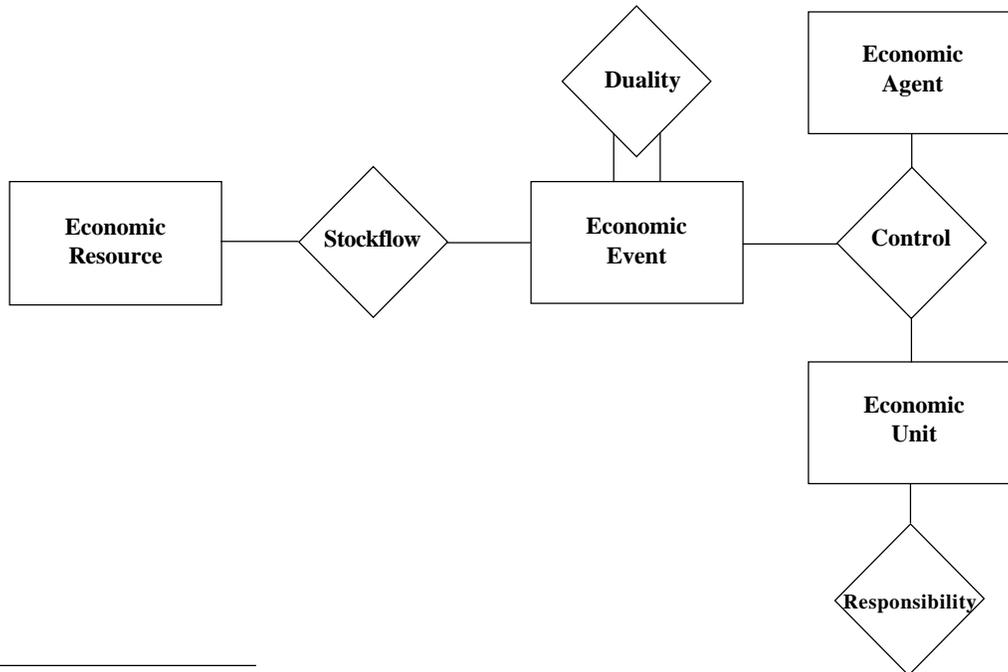
Section II of this article summarizes the REA/REAL model, and section III reviews basic concepts of data warehouse models. Section IV investigates multidimensional modeling using both star and snowflake schemas in the data warehouse setting, and section V relates the star and snowflake schemas to the REA/REAL schema and discusses some of the differences. Section VI summarizes the contributions and discusses potential extensions.

II. REA AND REAL MODELS

From an analysis of traditional account structures, McCarthy (1982) developed a generalized E-R (entity-relationship) model of accounting phenomena called the Resources, Events, and Agents (REA) accounting model. Agents are "persons and agencies who participate in the economic events of the enterprise or who are responsible for subordinates' participation" (McCarthy 1982, 563.) Economic units are "a subset of economic agents. Units are inside participants: agents who work for or are part of the enterprise being accounted for" (McCarthy 1982, 563.) Figure 2 illustrates the REA model.

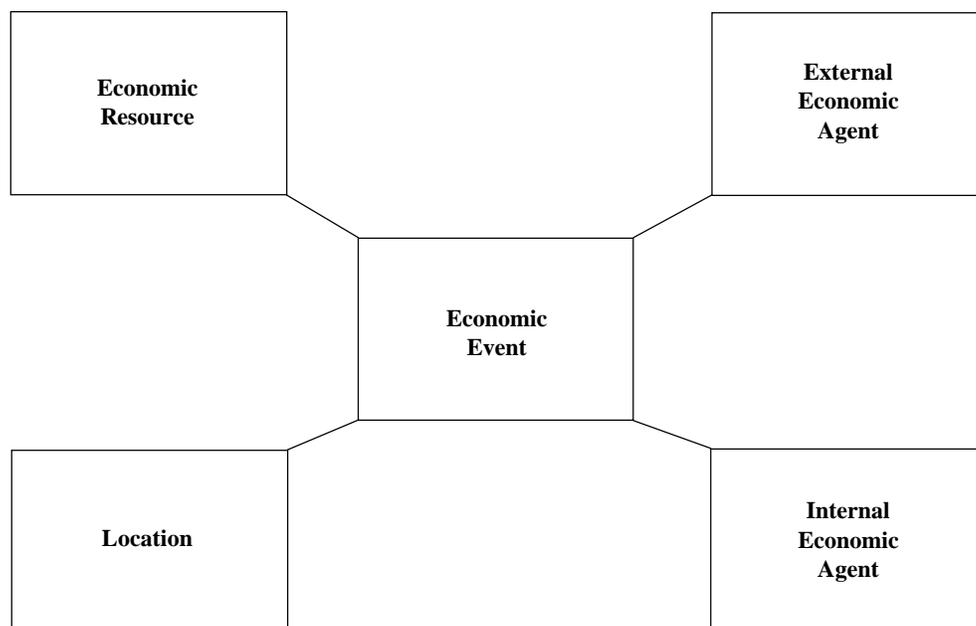
Denna et al. (1993) made the existence of the unit explicit by adding location, changing REA to REAL. The rationale was that if information about location is important, then unless it is derivable, it should be captured explicitly. Hollander et al. (1996) also note that location can be important for regulatory and other purposes. In addition, Hollander et al. (1996) give a location example where stores are given a unique identifier and information about the store is kept in a store table. Further, Hollander et al. (1996) differentiate between the internal and external agent. The resulting REAL model is illustrated in figure 3. This format is used here because diamonds are not used to capture relationships in the REAL or data warehouse literatures.

FIGURE 2
REA Model^a



^a Source: McCarthy (1982)

FIGURE 3
REAL Model



In addition, *time period* is an important part of the REA/REAL model. For example, when an event occurred and how long it took are captured in an REAL database (Hollander et al. 1996). Because time period is unique, Hollander et al. (1996) treat absolute time as an attribute. Subsequent research on REA/REAL has yielded an illustrative accounting system (Gal and McCarthy 1986) and REA for decision support (Denna and McCarthy 1987). In each case, the REA schema was developed to capture accounting and other transaction data in a manner that limited redundancy. When derivable, information such as location is not an explicit part of the REA schema.

Because REA is a theoretical ideal, there will be a range of implementation compromises necessitated by adoption of REA to various platforms. McCarthy and Rockwell (1989) noted examples of REA compromises that are a consequence of the implementation challenges: temporal aggregation of event histories, representation and use of a subset or super set of entities, and non-implementation of entity sets.

Because of these kinds of implementation compromises, the data warehouse may not be REA/REAL. It may not be optimal to include all customer data or all financial data in a data warehouse because it might not meet user needs (Lambert 1995). In addition, there can be extensions to the theoretical REA/REAL frameworks so that they meet different needs of data warehouse settings. Unique demands of data warehouses establish additional functions beyond those in the REA/REAL data schema, such as inclusion of a model of time periods in REAL-D.

III. DATA WAREHOUSES AND DATA MARTS

A data warehouse is “a single place located across a corporation’s networks where any user can get the latest data, efficiently organized” (Radding 1995, 57). A data warehouse is a(n):

1. subject-oriented (e.g., customer, vendor, product, etc.),
2. integrated (integration shows up in many different ways—in consistent naming conventions, in consistent measurement of variables, in consistent encoding structures, in consistent physical attributes of data, and so forth),
3. time-variant (over some time horizon), and
4. nonvolatile (not subject to update problems)

collection of data in support of management decision making (Inmon 1995). A data warehouse undergoes two processes: loading and access of data. Compared to transaction processing, there is no general update of data. As a result, liberties can be taken to optimize the access of data at the physical level of design (Inmon 1995).

Data marts are small data warehouses with specific business functions. For example, data marts have been used to facilitate measuring the impact of marketing promotions. Because data marts are developed for specific business purposes, system design, implementation, testing and installation are less costly than for data warehouses. Where data warehouses can take years to develop, data marts can be developed in a few months, at a much smaller cost.

Organizations use data warehouses for several objectives: to avoid slowing the performance of operational databases, to accumulate data over extended periods of time to allow analysis of trends, and to answer particular queries, such as how many units were sold in a month (Bull and Richman 1995). However, the most important reason is that they can be used to improve profits (e.g., *Public Accounting Report* 1997). Using data warehouses, firms can micro-manage product lines across time period and location, firms can create knowledge from data, and firms can perform a range of other activities. In addition, data warehouses have helped firms identify their most profitable customers (*Public Accounting Report* 1997).

In spite of the large size of data warehouses,³ users expect rapid response to queries. For example, a large regional retailer expects decision support responses within two minutes (Baer 1996), while

³ Some reported sizes for data warehouses are Holiday Inn’s 165 gigabyte warehouse for two years of historical data (DePompa 1996), Fingerhut Companies’ 300 gigabyte database (Baer 1996), Chase Manhattan Bank’s 560 gigabyte data warehouse (Baer 1996), MasterCard OnLine’s 1.2 terabyte database (Foley 1996), and Sears’ plan for a 2-terabyte data warehouse on 90 million households and 31 million Sears Card Holders (Foley 1996). Sears provides 121 weeks of daily sales history for each item by location to allow analysis by both time period and location (Greenberg 1996).

others do not want users to wait more than a few seconds. Consequently, the design of data warehouses must facilitate rapid query response. Accordingly, data warehouse systems are designed to respond to queries through multidimensional modeling and by using the star and snowflake schema.

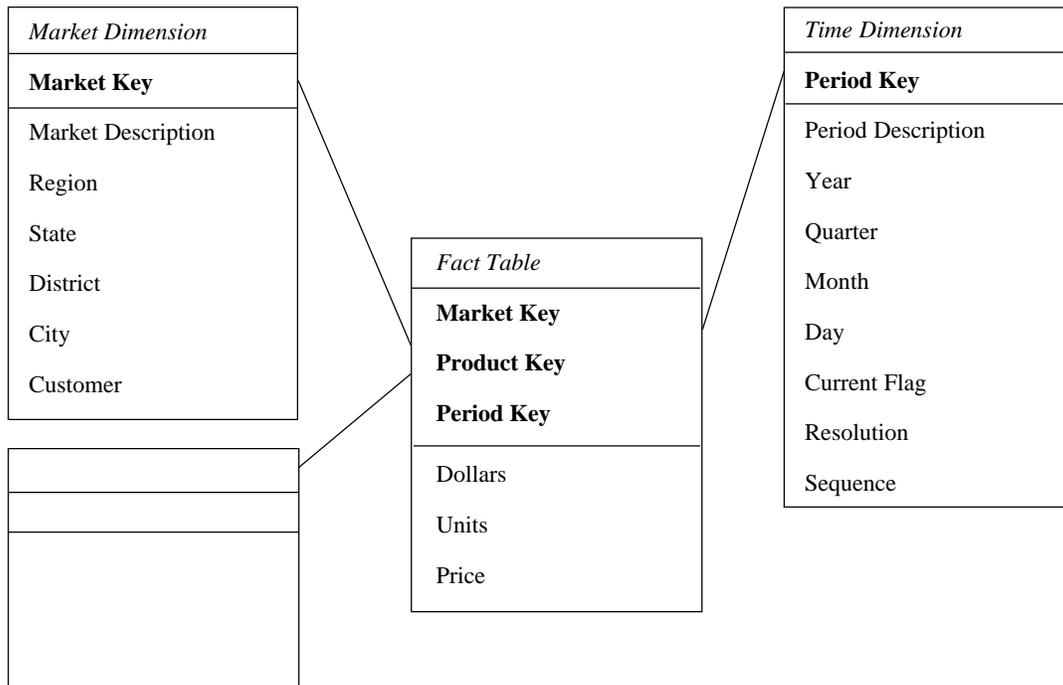
IV. MULTIDIMENSIONAL MODELING: STAR AND SNOWFLAKE SCHEMA

Multidimensional modeling (MDM), developed for online analytic processing, is “a technique for conceptualizing business models as a set of measures described by ordinary facets of business...based on facts, dimensions hierarchies and scarcity...designed around numeric data, such as values, counts, weights and occurrences” (Raden 1996a, 60). Data warehouses provide multidimensionality so that users can drill down to more detail in their queries. Multidimensional databases meet the needs of users that need to see data in a particular format, perform *ad hoc* analysis, and analyze the data using a range of techniques (Sahin 1995).

The primary difference between relational and multidimensional databases is that multidimensional databases store data in a format that facilitates users getting access to familiar data views and provide users the ability to drill down into the data. In multidimensional databases there is a hypercube, where, for each dimension on the hypercube, the user can drill down or aggregate. If the necessary data were present, relational databases could support virtual access to data views, but “[f]or these types of actions, a multidimensional database is both easier and faster to use than a relational database” (Sahin 1995, p. 2). A typical MDM problem statement is “What is the profitability by customer over time and organization?” (Raden 1996a, 61). Rather than focussing on transaction processing, MDM aims at answering managerial questions using the data. MDM capabilities are incorporated into relational databases, such as Oracle (e.g., Radding 1995). Schema design facilitates MDM.

The star schema employs two types of tables, *fact* tables at the center of the star and *dimension* tables at the points of the star. Fact tables capture event information and foreign key references to

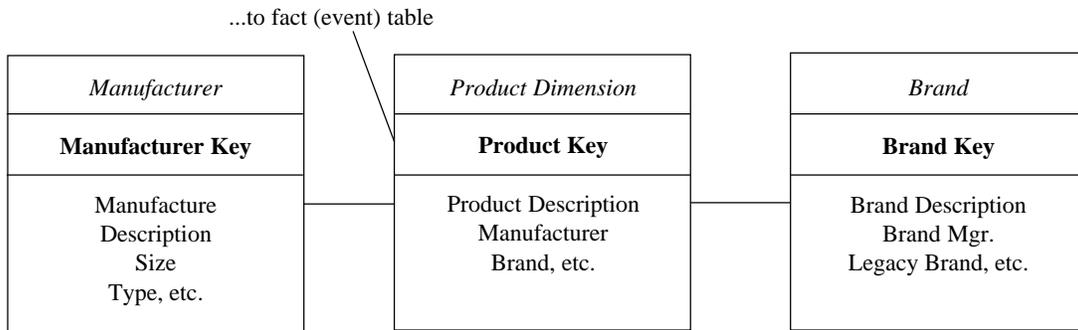
FIGURE 4
Star Schema^a



^a Based on Raden (1996a, 61).

dimension tables containing detail information. Dimension tables are kept on each dimension that decision makers would like to either rollup or drill down. In terms of REA/REAL, the event table is at the center of the star, and tables relating to the resources, agents, and locations are points of the star. The event table includes the foreign key field for each of the points on the star. Data warehouses can provide aggregated information or artifact-based information in star schema. For example, a data warehouse

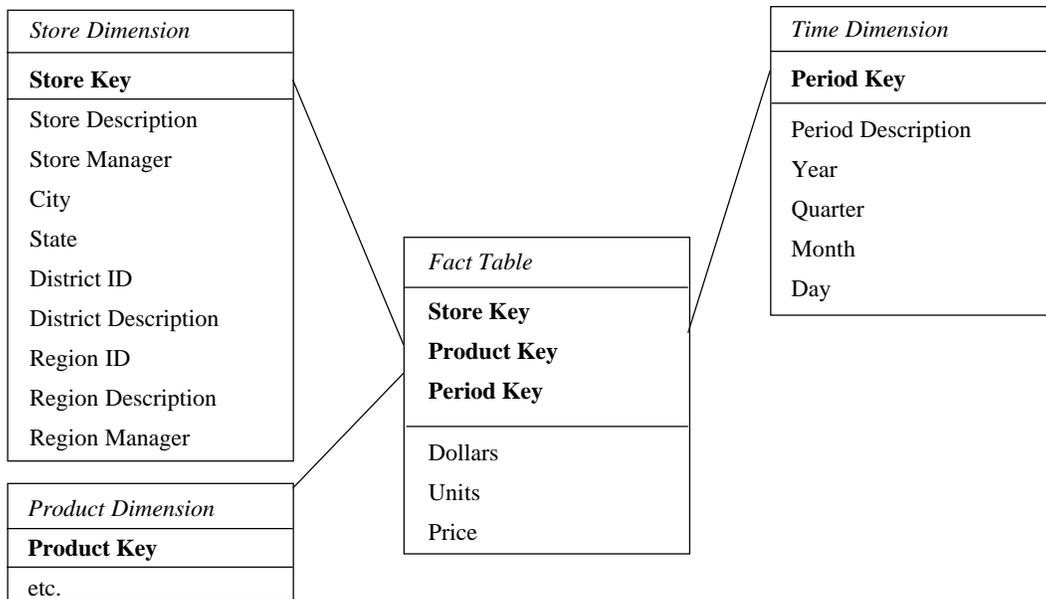
FIGURE 5
Turning a Star Schema (figure 2) into a Snowflake Schema^a



^a Based on Raden (1996a).

could facilitate analysis/aging of accounts receivable. Each receivable would be captured in the system allowing analysis across dimensions of time period, customer and location (e.g., salesperson, office and district). An example of the star schema, which appears throughout the data warehousing literature (e.g., Peterson 1995; Meredith and Khader 1996) is shown in figure 4.

FIGURE 6a
Star Schema^a



^a Based on Raden (1996b, 43).

FIGURE 6b
Sample Star Schema Data

Some Dimension Tables

<u>Store Key</u>	<u>Description</u>	<u>City</u>	<u>State</u>	<u>District</u>	<u>Region</u>
MM21	Mini Mall #21	01	05	01	01
MM22	Mini Mall #22	01	05	01	01

Product Key Product Description

aa2143 Wrist Band

<u>Period Key</u>	<u>Month</u>	<u>Date</u>	<u>Quarter</u>	<u>Year</u>
010198	01	01	01	1998
010298	01	02	01	1998

Event/Fact Table

<u>Store Key</u>	<u>Product Key</u>	<u>Period Key</u>	<u>Dollars</u>	<u>Units</u>	<u>Price</u>
MM21	aa2143	010198	100	5	20

When a database includes a large number of dimensions (categories) and dimensions are large (high cardinality), Raden (1996a) recommends using the snowflake schema, which creates tables for attributes within a dimension table. A snowflake is similar to a star except that foreign keys can be embedded in the dimensions so that dimension tables have relationships with other dimension tables. An example snowflake model extension to figure 4 appears in figure 5.

Another example of a star is given in figure 6a, and sample data for that file are provided in figure 6b. Data warehouses need information that explicitly captures all dimensions that users wish to summarize. For example, quarter is derivable from date, but data warehouses would typically include quarter so that user requests for quarter data could be answered rapidly.

Use of the star or snowflake schemas is aimed at limiting access and query problems in a data warehouse environment. As Raden (1996a) noted, dimensional modeling for relational database design exploits creation of long thin fact tables containing numeric information and relatively short and wide dimension tables. With this approach, a number of queries can be resolved without reading the fact table, thus avoiding time-consuming table scans and complex joins.

The star schema occurs often in commercial software. For example, one of Red Brick Warehouse's distinctive features is its ability to join star schema data rapidly in response to queries (Edelstein 1995). Bontempo and Saracco (1997) discuss approaches to improve processing speed for star schema using bitmaps. Because of the read nature of data warehousing, overhead can be reduced with a sufficient amount of write cache (Raden and Peterson 1997). Star schema can exhibit unanticipated overhead. For example, precalculated data generated in anticipation of particular queries for star and snowflake schema add to the complexity of the data warehouse and result in more indexes (and associated reindexing) (Watterson 1997). Aggregated tables and indexes to the fact and aggregate fact tables consume substantially more space than raw data (Greenfield 1996).

V. STAR AND SNOWFLAKE MODELS AND REA/REAL

Schema Similarities and Differences

The star and snowflake models in data warehouses are structurally similar to REA/REAL models. Because data warehouses focus on anticipating queries on resources and agents, data warehouse star and snowflake schemas include information on these dimensions, just as in REA/REAL. Resources in REA/REAL are also resources in the star and snowflake models used for data warehousing

and MDM. For example, in figure 4, the product dimension is a resource. Agents in the REA are also agents in MDM.

Data warehouses require unique and different information than the REA/REAL schema. First, time period is captured as a dimension rather than in a single attribute for time as in REA/REAL models (McCarthy 1979, 1980, 1982; Denna et al. 1993; Hollander et al. 1996). Second, a wider range of information about location is captured, e.g., store, city, region and district. Third, some dimensions are not homogeneous as they are in REA/REAL models, in that agents are mixed with locations (e.g., figure 4). Fourth, the concern of the REA model with economic unit was subordinated to agent and was one of control, whereas the primary concern of data warehouses is one of marketing information. The remainder of this section addresses each of these differences.

Time Period as a Dimension

In figure 4, an entire dimension defines time periods. The existence of this data as a dimension does not necessarily increase data entry demands. For example, some of data in the dimensions will be generated automatically such as calendar conversions for day to week to month to quarter to year, which lets rollups be automatic (Raden 1996a). In addition, special-case time ranges, including promotion periods and seasons, can be built into the database. As a result, REA-based databases could form the basis of replicated databases in data warehouses, where the replication automatically expands the available data to a broader based schema as in figure 4.

Location as a Dimension

In data warehouses, location (e.g., market) is often a dimension. Although Denna et al. (1993) and Hollander et al. (1996) suggest that if location can be inferred then information about it does not need to be captured, the size of the data warehouse and the need for rapid query response argue for explicit rather than inferred information to facilitate rollups, even if the information is redundant or derivable. Location information can be generated automatically, in the same manner as time information. As a result, if store numbers or register numbers are unique, they can be used to generate location information.

Homogeneity of Dimensions in REA and Data Warehouse Models

A schema table for a dimension (resource, event, agent, location) is defined as *totally homogeneous* if that table contains *only* information directly relating to that dimension (e.g., resources) and not any other dimensions (e.g., agents). While the REA/REAL literature illustrates resources, agents, and locations as totally homogeneous, under the data warehouse formulation the dimension tables are not totally homogeneous according to the same criteria. For example, in figure 6a the dimension for store key contains information about agents and locations. Why does the data warehouse version not maintain resource, agent, and location homogeneity? In these examples (e.g., Meredith and Khader 1996), there is *direct* concern for being able to answer marketing queries directly (e.g., about the sales organization marketing representative, office, district and region). Therefore, the data warehouse design schema concern is not with the homogeneity of agent or location information but instead with *cumulating nonhomogeneity*, which, like time, cascades into definable categories. For example, the salesperson is the lowest level in sales, all the sales personnel in a store have sales that accumulate to the store sales, all the store sales accumulate to the district sales, all the district sales accumulate to all the region sales, etc. Consequently, a dimension has cumulating nonhomogeneity if for some resource, agent, or location, data are included for more than one dimension (such as sales personnel or store location).

Incorporating cumulating nonhomogeneity in the schema minimizes the number of physical joins that must be made and provides a simpler schema (e.g., Raden 1996a). In the context of drill down for location and time breakdowns for very large databases, minimizing physical joins can be a critical design objective. The notion of cumulating nonhomogeneity can be extended to other dimensions related to other events such as purchasing.

On the one hand, employing cumulating nonhomogeneity could be viewed as an implementation compromise because it intermixes information, e.g., agent and location information (McCarthy and

Rockwell 1989). On the other hand, it could be argued that cumulating nonhomogeneity provides a different perspective on the system design. Instead of focusing on the origin of the data, cumulating nonhomogeneity focuses on the goal of providing answers to questions about organizational activities, such as creating value. Development of similar models from different perspectives provides evidence of the robustness of the REA/REAL approach. Cumulative nonhomogeneity in data warehouses typically results from mixing either external agents (figure 4) and locations or internal agents and locations (figure 6a). The mixing that occurs depends on the type of questions the data warehouse is designed to answer.

Control vs. Marketing Questions

In McCarthy (1982), economic unit is related to the event through the relationship control. Because data warehouses support a range of queries related to creating value (e.g., Appleton 1996), there is likely to be concern for additional types of relationships, e.g., marketing. Sales and customer information is the most important data in many data warehouses. For example, Sears chose data warehousing “to be the single most authoritative source for sales information” (Greenberg 1996, 66). Holiday Inn’s data warehouse is used “to look at stay patterns of our customers...who stayed with us, when for how long [and] how they paid” (DePompa 1996). Another use is to identify the most profitable customers (*Public Accounting Report* 1997).

Although transaction-gathering schema and accounting information systems provide substantial attention to control issues, data warehouses are concerned with creating value and generating profits. As a result, because the transactions have already been through the control system, there is less interest in control and more interest in analysis of sales data in order to improve profits and create value.

FIGURE 7
REAL-D Model

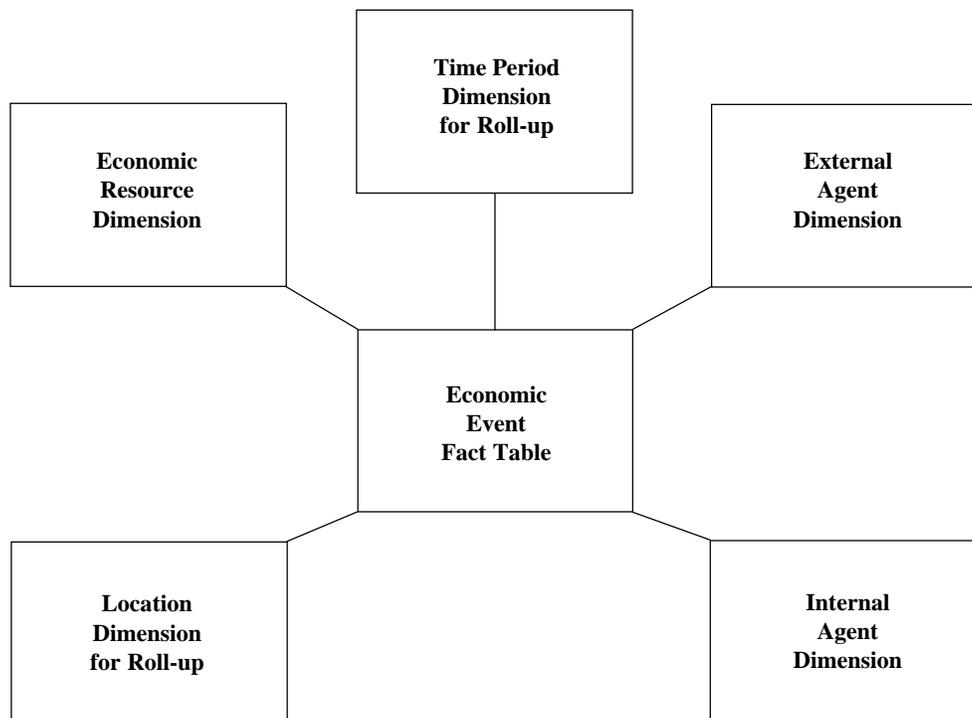


FIGURE 8
Sample Star Schema REAL-D

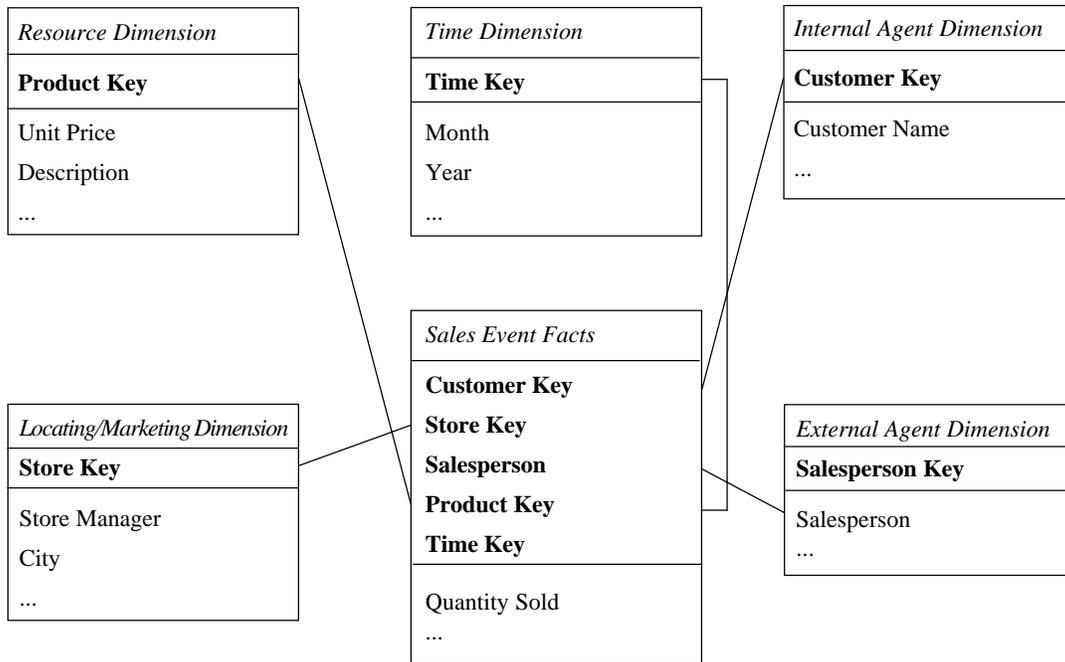
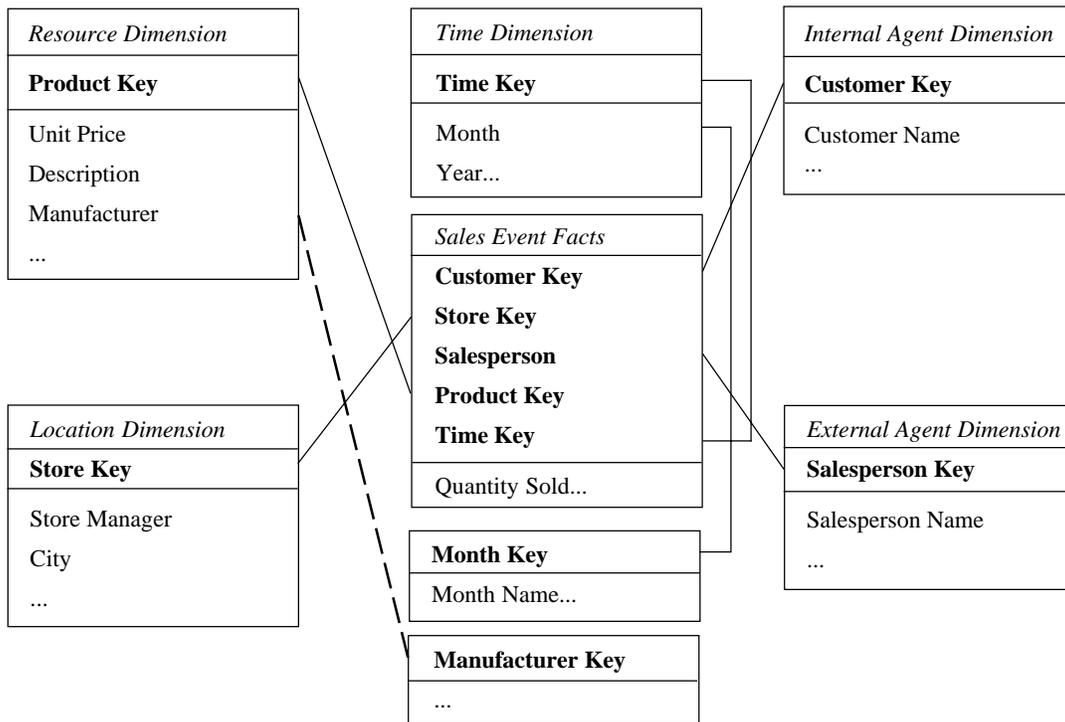


FIGURE 9
Sample Snowflake Schema for REAL-D



Accordingly, designers focus on functional needs, changing the focus of data warehouses to market information.

A Data Warehouse Model

REAL-D, the revised model, can be represented with a star or a snowflake schema. At the center of the star is the event table, capturing the critical information about the event (the fact table). Surrounding the event are the dimensions of resources, agents, location, and time period as they relate to the event, resulting in a star schema. A general example appears in figure 7; a particular example, in figure 8. In some situations, however, additional tables will need to be generated from some of the dimensions, resulting in a snowflake schema. In the example in figure 9, additional tables are included to provide information about time period and manufacturer. As in REA and REAL models, the particular process being modeled influences which resources, events, agents, and locations are included and the number of tables used to represent each. Additional information changes the star model into a snowflake model. Control information can also be added with links from agents to economic unit, or agent information can be linked to resource information.

Implementation

Data warehouse star schema systems can be implemented from REA/REAL databases by mapping the original database into a star schema using SQL. Alternatively, some tools have been designed to help developers generate a hypercube based on a relational database (DePompa 1997). At the center of the star would be the event table, which holds the particular measures of the event. Then the points of the star would be tables containing the resources, agent, and location dimensions. Further, the administrator would add any necessary computed columns for aggregations or unique OLAP use. In addition, the database administrator would also create the necessary timetable with the appropriate date hierarchy. From an efficiency standpoint, it is often helpful to order the data by the most prevalent use, which minimizes sorting to respond to queries (Petersen 1994). As part of the implementation, any compromises to the original REA/REAL structure, such as nonrepresentation of any entity sets, would be generated.

VI. DISCUSSION

This article integrated REA/REAL models developed for capturing event information and data warehouse requirements for decision making in an integrated schema called REAL-D. The resulting schema incorporates explicit time period and location information into the event model of REA/REAL. This research has expanded REA/REAL to model data warehouses, a setting not previously investigated by REA/REAL researchers.

Dunn and McCarthy (1997, 46) suggested that "productive extensions in REA research could include...use of REA to explicate instantiations." New developments in data warehouses could be used to generate additional extensions to the REA/REAL model. First, as noted above, the sales event is not the only event likely to have different data warehouse dimensions. The purchase event has the same structure. For example, firms purchase from a set of sales people that work for the same division of a firm that is part of a larger organization. The purchases accumulate to purchases from a division, which accumulate to purchases from a company. This is important information because a decision maker can use it to generate purchase information for negotiating quantity discounts from sellers. Second, a behavioral study of data warehouse designers could be run to determine whether designers trained in REAL-D outperform those without training. Providing researchers a template for analyzing data warehouse design is likely to improve their results. Third, a study could compare data warehouses for their similarity to REAL-D.

APPENDIX
Some Useful URLs for Data Warehousing Schema

Content	Uniform Resource Locator (URL)
CIO Data Warehousing Links	http://www.cio.com/CIO/rc_dw.html
Data Warehousing Information	http://pwp.starnetinc.com/larryg/index.html
Article List	http://pwp.starnetinc.com/larryg/articles.html
White Article List	http://pwp.starnetinc.com/larryg/whitepap.html
Data Warehousing Institute	http://www.dw-institute.com/
Best of Database: Programming and Design	http://www.dbpd.com/vault/index.shtml
Association for Computing Machinery's Special Interest Group on Management of Data	http://www.acm.org/sigmod/
Stanford Data Warehousing Publications	http://www-db.stanford.edu/warehousing/publications.html

Content	Uniform Resource Locator (URL)
Custom Net	http://www.custom-net.com/dw.htm
Lycos Search for Data Warehouse	http://query6.lycos.cs.cmu.edu/cgi-bin/pursuit?query=data+wareh

Some Useful URLs for OLAP

Content	Uniform Resource Locator (URL)
OLAP Resources	http://www.sgroves.demon.co.uk/olapltnks.htm
Lycos Search on OLAP	http://lycos11.lycos.cs.cmu.edu/cgi-bin/pursuit?query=olap
Maintenance and control of OLAP applications	http://www.xs4all.nl/~fab/olapkeep.html
Relational OLAP Systems	http://www.cas.american.edu/~ghadsal/DW/OLAP/sld003.htm
OLAP and the Web	http://www.dbmsmag.com/9701i08.html
OLAP, OLTP, MDDB	http://www.datamation.com/PlugIn/issues/1996/april15/04beval1.html

REFERENCES

- Appleton, E. 1996. Use your data warehouse to compete. *Datamation* (May 15): 34–38.
- Baer, T. 1996. Relational technology in the land of the giants. *Software Magazine* (February): 61–64.
- Bontempo, C., and C. Saracco. 1997. Accelerating indexed searching. *Database Programming and Design Online*. <<http://www.dbpd.com/vault/bontempo.htm>>.
- Bull, K., and D. Richman. 1995. Oracle's single vendor solution. *Information Week* (July 10): 72.
- Darling, C. 1996. How to integrate your data warehouse. *Datamation* (May 15) 40–52.
- Denna, E., J. Cherrington, D. Andros, and A. Hollander. 1993. *Event Driven Business Solutions*. Homewood, IL: Irwin.
- , and W. McCarthy. 1987. An events accounting foundation for DSS. In *Decision Support Systems: Theory and Application*, edited by C. Holsapple, and A. Whinston, 239–263. Berlin, Germany: Springer Verlag.
- DePompa, B. 1996. Stack that data. *Information Week* (January 29): 50–57.
- . 1997. How to sort through the OLAP maze. *DW for Data Warehousing* (May).
- Dunn, C., and W. McCarthy. 1997. The REA accounting model: Intellectual heritage and prospects for progress. *Journal of Information Systems* (Spring): 31–51.

- Edelstein, H. 1995. Technology analysis: Faster data warehouses: New tools provide high-performance querying through advanced indexing. *Information Week* (December 4): <<http://www.techweb.com/se/directlink.cgi?IWK19951204S0045.htm>>.
- Foley, J. 1996. Data dilemma. *Information Week* (June 10): 14–16.
- Gal, G., and W. McCarthy. 1986. Operation of a relational accounting system. *Advances in Accounting*: 83–112.
- Greenberg, I. 1996. Data warehouse initiative helps Sears weather competition. *Infoworld* (July 29): 66.
- Greenfield, L. 1996. Don't let data warehousing gotchas getcha. *Datamation* (March 1): <<http://www.datamation.com/PlugIn/issues/1996/march1/03asoft1.html>>.
- Hollander, A., E. Denna, and J. Cherrington. 1996. *Accounting, Information Technology, and Business Solutions*. Chicago, IL: Irwin.
- Inmon, W. 1995. *What is a Data Warehouse?* <http://www.cait.wustl.edu/cait/papers/prism/vol1_no1/>.
- Lambert, B. 1995. Break old habits to define data warehousing requirements. *Data Management Review* (December): <<http://www.data-warehouse.com/resource/articles/lamber11.htm>>.
- McCarthy, W. 1979. An entity view of accounting models. *The Accounting Review* (October): 667–685.
- . 1980. Construction and use of integrated accounting systems with entity relationship modeling. In *Entity-Relationship Approach to Systems Analysis and Design*, edited by P. Chen, 625–650. New York, NY: North-Holland.
- . 1982. The REA accounting model: A generalized framework for accounting systems in a shared data environment. *The Accounting Review* (July): 554–578.
- , and S. Rockwell. 1989. The integrated use of first-order theories, reconstructive expertise and implementation heuristics in an accounting information system design tool. *Proceedings of the Ninth International Workshop on Expert Systems and their Applications*, Avignon, France, EC2.
- Meredith, M., and A. Khader. 1996. Divide and aggregate: Designing large data warehouses. *Database Programming and Design: The Online Edition* (June): <<http://www.dbpd.com/vault/archives.shtml>>; <<http://www.dbpd.com/vault/96index.htm>>.
- Paul, L. 1997. Anatomy of a failure. *CIO Magazine* (November 15): <http://www.cio.com/archive/enterprise/111597_data.html>.
- Petersen. 1994.
- Peterson, S. 1995. Stars: A pattern language for query optimized schema. <<http://c2.com/ppr/stars.html>>.
- Public Accounting Report*. 1997. KPMG jumps in data warehousing ring. *Public Accounting Report* (January): 1, 5.
- Radding, A. 1995. Building a better data warehouse. *Infoworld* (November 20): 1, 57–62.
- Raden, N. 1996a. Modeling a data warehouse. *Information Week* (January 29): 60–66.
- . 1996b. Maximizing your warehouse. *Information Week* (March 18): 42–48.
- , and Peterson. 1997.
- Sahin, K. 1995. Multidimensional database technology and data warehousing. *Database Journal* (December).
- SAS. (date). *Data Warehousing: A Blueprint for Success*. Cary, NC: SAS Institute.
- Watterson, K. 1997. Parallel tracks. *Datamation* (May): <<http://www.datamation.com/PlugIn/issues/1997/may/05data.html>>.