Comp. by: SIndumathiGalleys0000844939 Date:14/6/08 Time:00:21:26 Stage:First Proof File Path://
ppdys1108/Womat3/Production/PRODENV/0000000005/0000008302/0000000016/0000844939.3D
Proof by: QC by:

A

Active and Real-Time Data Warehousing

Mukesh Mohania[1], Ullas Nambiar[1], Michael Schrefl[2], Millist Vincent[3]
[1]IBM India Research Lab, New Delhi, India
[2]University of Linz, Austria
[3]School of Computer and Information Science, University of South Australia, SA, Australia

Synonyms

Right-time data warehousing

Definition

Active Data Warehousing is the technical ability to capture transactions when they change, and integrate them into the warehouse, along with maintaining batch or scheduled cycle refreshes. An active data warehouse offers the possibility of automating routine tasks and decisions. The active data warehouse exports decisions automatically to the On-Line Transaction Processing (OLTP) systems.

Real-time Data Warehousing describes a system that reflects the state of the warehouse in real time. If a query is run against the real-time data warehouse to understand a particular facet about the business or entity described by the warehouse, the answer reflects the state of that entity at the time the query was run. Most data warehouses have data that is highly latent – or reflects the business at a point in the past. A real-time data warehouse has low latency data and provides current (or real-time) data.

Simply put, a real-time data warehouse can be built using an active data warehouse with a very low latency constraint added to it. An alternate view is to consider active data warehousing as being a design methodology suited to tactical decision-making based on very current data while real-time data warehousing is a collection of technologies that refresh a data warehouse frequently. A real-time data warehouse is one that acquires, cleanses, transforms, stores, and disseminates information in real time. An active data warehouse, on the other hand, operates in a non-real-time response mode with one-or-more OLTP systems.

Historical Background

A *data warehouse* is a decision support database that is periodically updated by extracting, transforming, and loading operational data from several OLTP databases. In the data warehouse, OLTP data is arranged using the (multi) dimensional data modeling approach (see [1] for a basic approach and [2] for details on translating an OLTP data model into a dimensional model), which classifies data into *measures* and *dimensions*. In recent years, several multidimensional data models have been proposed [3–6]. An indepth comparison is provided by Pedersen and Jensen in [5]. The basic unit of interest in a data warehouse is a *measure* or *fact* (e.g., sales), which represent countable, semisummable, or summable information concerning a business process. An instance of a measure is called *measure value*. A measure can be analyzed from different perspectives, which are called the *dimensions* (e.g., location, product, time) of the data warehouse [7]. A dimension consists of a set of *dimension levels* (e.g., time: Day, Week, Month, Quarter, Season, Year, ALLTimes), which are organized in multiple hierarchies or *dimension paths* [6] (e.g., Time[Day] → Time[Month] → Time[Quarter] → Time[Year] → Time[ALLTimes]; Time[Day] → Time [Week] → Time[Season] → Time[ALLTimes]). The hierarchies of a dimension form a lattice having at least one top dimension level and one bottom dimension level. The measures that can be analyzed by the same set of dimensions are described by a *base cube* or *fact table*. A base cube uses level instances of the lowest dimension levels of each of its dimensions to identify a measure value. The relationship between a set of measure values and the set of identifying level instances is called *cell*. Loading data into the data warehouse means that new *cells* will be added to *base cubes* and new level instances will be added to *dimension levels*. If a

Comp. by: SIndumathiGalleys0000844939 Date:14/6/08 Time:00:21:26 Stage:First Proof File Path://
ppdys1108/Womat3/Production/PRODENV/0000000005/0000008302/0000000016/0000844939.3D
Proof by: QC by:

dimension *D* is related to a measure *m* by means of a base cube, then the hierarchies of *D* can be used to aggregate the *measure values* of *m* using operators like SUM, COUNT, or AVG. Aggregating measure values along the hierarchies of different dimensions (i.e., *rollup*) creates a multidimensional view on data, which is known as *data cube* or *cube*. Deaggregating the measures of a cube to a lower dimension level (i.e.,*drilldown*) creates a more detailed cube. Selecting the subset of a cube's cells that satisfy a certain selection condition (i.e., *slicing*) also creates a more detailed cube.

The data warehouses are used by analysts to find solutions for decision tasks by using OLAP (On-Line Analytical Processing) [7] systems. The decision tasks can be split into three, viz. *non-routine, semi-routine*, and *routine*. Non-routine tasks occur infrequently and/or do not have a generally accepted decision criteria. For example, strategic business decisions like introducing a new brand or changing an existing business policy are non-routine tasks. Routine tasks on the other hand are well structured problems for which generally accepted procedures exist and they occur frequently and at predictive intervals. Examples can be found in the areas of product assortment (change price, withdraw product, etc.), customer relationship management (grant loyalty discounts etc.), and in many administrative areas (accept/reject paper based on review scores). Semi-routine tasks are tasks that require a non-routine solution – e.g., paper rated contradictory must be discussed by program committee. Since, most tasks are likely to be routine, it is logical to automate processing of such tasks to reduce the delay in decision-making.

Active data warehouses [8] were designed to enable data warehouses to support automatic decision-making when faced with routine decision tasks and routinizable elements of semi-routine decision tasks. The active data warehouse design extends the technology behind active database systems. Active database technology transforms passive database systems into reactive systems that respond to database and external events through the use of rule processing features [9,10]. Limited versions of active rules exist in commercial database products [11,12].

Real-time data warehousing captures business activity data as it occurs. As soon as the business activity is complete and there is data about it, the completed activity data flows into the data warehouse and becomes available instantly. In other words, real-time data warehousing is a framework for deriving information from data as the data becomes available. Traditionally, data warehouses were regarded as an environment for analyzing historic data, either to understand what has happened or simply to log the changes as they happened. However, of late, businesses want to use them to predict the future: e.g., to predict customers likely to churn; and thereby seek better control of the business. However, until recently, it was not practical to have *zero-latency* data warehouses – the process of extracting data had too much of an impact on the source systems concerned, and the various steps needed to cleanse and transform the data required multiple temporary tables and took several hours to run. However, the increased visibility of (the value of) warehouse data, and the take-up by a wider audience within the organization, has lead to a number of product developments by IBM [13], Oracle [14], and other vendors that make real-time data warehousing now possible.

Scientific Fundamentals

The two example scenarios below describe typical situations in which active rules can be used to automate decision-making: *Scenario 1 : Reducing the price of an article*. Twenty days after a soft drink has been launched on a market, analysts compare the quantities sold during this period with a standardized indicator. This indicator requires the total quantities sold during the 20-day period, not to drop below a threshold of 10,000 sold items. If the analyzed sales figures are below this threshold, the price of the newly launched soft drink will be reduced by 15.

Scenario 2 : Withdrawing articles from a market. At the end of every quarter, high-priced soft drinks that are sold in Upper Austrian stores will be analyzed. If the sales figures of a high-priced soft drink have continuously dropped, the article will be withdrawn from the Upper Austrian market. Analysts inspect sales figures at different granularities of the time dimension and at different granularities of the location dimension. Trend, average, and variance measures are used as indicators in decision-making.

Rules that mimic the analytical work of a business analyst are called *analysis rules* [8]. The components of analysis rules constitute the *knowledge model* of an active data warehouse (and also a real-time data warehouse). The knowledge model determines *what* an

Comp. by: SIndumathiGalleys0000844939 Date:14/6/08 Time:00:21:27 Stage:First Proof File Path://
ppdys1108/Womat3/Production/PRODENV/0000000005/0000008302/0000000016/0000844939.3D
Proof by: QC by:

analyst must consider, when he specifies an active rule to automate a routine decision task.

An analysis rule consists of (i) the *primary dimension level* and (ii) the *primary condition*, which identify the objects for which decision-making is necessary, (iii) the *event*, which triggers rule processing, (iv) the *analysis graph*, which specifies the cubes for analysis, (v) the *decision steps*, which represent the conditions under which a decision can be made, and (vi) the *action*, which represents the rule's decision task. Below is a brief description of the components of an analysis rule. Detailed discussion is given in [8].

Event: Events are used to specify the timepoints at which analysis rules should be carried out. Active data warehouses provide three kinds of events: (i) OLTP method events, (ii) relative temporal events, and (iii) calendar events. OLTP method events describe basic happenings in the data warehouse's sources. Relative temporal events are used to define a temporal distance between such a basic happening and carrying out an analysis rule. Calendar events represent fixed points in time at which an analysis rule may be carried out. Structurally, every event instance is characterized by an occurrence time and by an event identifier. In its event part, an analysis rule refers to a calendar event or to a relative temporal event.

An *OLTP method event* describes a happening in the data warehouse's source systems that is of interest to analysis rules in the active data warehouse. Besides occurrence time and event identifier, the attributes of an OLTP method event are a reference to the dimension level for which the OLTP method event occurred and the parameters of the method invocation. To make OLTP method events available in data warehouses, a data warehouse designer has to define the schema of OLTP method events and extend the data warehouse's extract/transform/load mechanism. Since instances of OLTP method events are loaded some time after their occurrence, analysis rules cannot be triggered directly by OLTP method events.

Temporal events determine the timepoints at which decision-making has to be initiated. Scenario 1 uses the relative temporal event "twenty days after launch" while Scenario 2 uses the periodic temporal event "end of quarter." The *conditions* for decision-making are based on indicators, which have been established in manual decision-making. Each condition refers to a multidimensional cube and therefore "analyzing" means to evaluate the condition on this cube. Scenario 1 uses a quantity-based indicator, whereas scenario 2 uses value-based indicators for decision-making. The decision whether to carry out the rule's *action* depends on the result of evaluating the conditions. The action of scenario 1 is to reduce the price of an article, whereas the action of scenario 2 is to withdraw an article from a market.

Primary Condition: Several analysis rules may share the same OLTP method as their action. These rules may be carried out at different timepoints and may utilize different multidimensional analyses. Thus, a certain analysis rule usually analyses only a subset of the level instances that belong to the rule's primary dimension level. The primary condition is used to determine for a level instance of the primary dimension level whether multidimensional analysis should be carried out by the analysis rule. The primary condition is specified as a Boolean expression, which refers to the describing attributes of the primary dimension level. If omitted, the primary condition evaluates to TRUE.

Action: The purpose of an analysis rule is to automate decision-making for objects that are available in OLTP systems and in the data warehouse. A *decision* means to invoke (or not to invoke) a method on a certain object in an OLTP system. In its action part, an analysis rule may refer to a single OLTP method of the primary dimension level, which represents a transaction in an OLTP system. These methods represent the *decision space* of an active data warehouse. To make the transactional behavior of an OLTP object type available in the active data warehouse, the data warehouse designer must provide (i) the specifications of the OLTP object type's methods together with required parameters, (ii) the preconditions that have to be satisfied before the OLTP method can be invoked in the OLTP system, and (iii) a conflict resolution mechanism, which solves contradictory decisions of different analysis rules. Since different analysis rules can make a decision for the same level instance of the rules' primary dimension level during the same active data warehouse cycle, a *decision conflict* may occur. Such conflicts are considered as interrule conflicts. To detect interrule conflicts, a *conflict table* covering the OLTP methods of the decision space is used. The tuples of the conflict table have the form $<m1, m2, m3>$, where $m1$ and $m2$ identify two conflicting methods and $m3$ specifies the conflict resolution method that will be finally

Comp. by: SIndumathiGalleys0000844939 Date:14/6/08 Time:00:21:27 Stage:First Proof File Path:// ppdys1108/Womat3/Production/PRODENV/0000000005/0000008302/0000000016/0000844939.3D
Proof by: QC by:

executed in OLTP systems. If a conflict cannot be solved automatically it has to be reported to analysts for manual conflict resolution.

Analysis Graph: When an analyst queries the data warehouse to make a decision, he or she follows an incremental topdown approach in creating and analyzing cubes. Analysis rules follow the same approach. To automate decision-making, an analysis rule must "know" the cubes that are needed for multidimensional analysis. These cubes constitute the *analysis graph*, which is specified once by the analyst. The n dimensions of each cube of the analysis graph are classified into one *primary dimension*, which represents the level instances of the primary dimension level, and $n - 1$ *analysis dimensions*, which represent the multidimensional space for analysis. Since a level instance of the primary dimension level is described by one or more cells of a cube, multidimensional analysis means to compare, aggregate, transform, etc., the measure values of these cells. Two kinds of multidimensional analysis are carried out at each cube of the analysis graph: (i) select the level instances of the primary dimension level whose cells comply with the decision-making condition (e.g., withdraw an article if the sales total of the last quarter is below USD 10,000) and (ii) select the level instances of the primary dimension level whose cells comply with the condition under which more detailed analysis (at finer grained cubes) are necessary (e.g., continue analysis if the sales total of the last quarter is below USD 500,000). The multidimensional analysis that are carried out on the cubes of the analysis graph are called decision steps. Each decision step analyzes the data of exactly one cube of the analysis graph. Hence, analysis graph and decision steps represent the knowledge for multidimensional analysis and decision-making of an analysis rule.

Enabling real-time data warehousing: As mentioned earlier, real-time data warehouses are active data warehouses that are loaded with data having (near) zero latency. Data warehouse vendors have used multiple approaches such as *hand-coded scripting* and data extraction, transformation, and loading (ETL) [15] solutions to serve the data acquisition needs of a data warehouse. However, as users move toward real-time data warehousing, there is a limited choice of technologies that facilitate real-time data delivery. The challenge is to determine the right technology approach or combination of solutions that best meets the data delivery needs. Selection criteria should include considerations for frequency of data, acceptable latency, data volumes, data integrity, transformation requirements and processing overhead. To solve the real-time challenge, businesses are turning to technologies like enterprise application integration (EAI) [16] and transactional data management (TDM) [17], which offers high-performance, low impact movement of data, even at large volumes with sub-second speed. EAI has a greater implementation complexity and cost of maintenance, and handles smaller volumes of data. TDM provides the ability to capture transactions from OLTP systems, apply mapping, filtering, and basic transformations and delivers to the data warehouse directly. A more detailed study of the challenges involved in implementing a real-time data warehouse is given in [18].

Key Applications

Active and Real-time data warehouses enable businesses across all industry verticals to gain competitive advantage by allowing them to run *analytics* solutions over the most recent data of interest that is captured in the warehouse. This will provide them with the ability to make intelligent business decisions and better understand and predict customer and business trends based on accurate, up-to-the-second data. By introducing real-time flows of information to data warehouses, companies can increase supply chain visibility, gain a complete view of business performance, and increase service levels, ultimately increasing customer retention and brand value.

The following are just some of the other business benefits of active and real-time data warehousing:

- *Real-time Analytics:* Real-time analytics is the ability to use all available data to improve performance and quality of service, at the moment they are required. It consists of dynamic analysis and reporting, right at the moment (or very soon after) the resource (or information) entered the system. In a practical sense, real time is defined by the need of the consumer (business) and can vary from a few seconds to few minutes. In other words, more frequent than daily can be considered real-time, because it crosses the overnight-update barrier. With increasing availability of active and real-time data warehouses, the technology for capturing and analyzing real-time data is increasingly becoming available. Learning how to apply it effectively

Comp. by: SIndumathiGalleys0000844939 Date:14/6/08 Time:00:21:27 Stage:First Proof File Path://
ppdys1108/Womat3/Production/PRODENV/0000000005/0000008302/0000000016/0000844939.3D
Proof by: QC by:

becomes the differentiator. Implementing real-time analytics requires the integration of a number of technologies that are not interoperable off-the-shelf. There are no established best practises. Early detection of fraudulent activity in financial transactions is a potential environment for applying real-time analytics. For example, credit card companies monitor transactions and activate counter measures when a customer's credit transactions fall outside the range of expected patterns. However, being able to correctly identify fraud while not offending a well-intentioned valuable customer is a critical necessity that adds complexities to the potential solution.

- *Maximize ERP Investments:* With a real-time data warehouse in place, companies can maximize their Enterprise Resource Planning (ERP) technology investment by turning integrated data into business intelligence. ETL solutions act as an integral bridge between ERP systems that collect high volumes of transactions and business analytics to create data reports.
- *Increase Supply Chain Visibility:* Real-time data warehousing helps streamline supply chains through highly effective business-to-business communications and identifies any weak links or bottlenecks, enabling companies to enhance service levels and gain a competitive edge.
- *Live 360° View of Customers:* The active database solutions enable companies to capture, transform, and flow all types of customer data into a data warehouse, creating one seamless database that provides a 360° view of the customer. By tracking and analyzing all modes of interaction with a customer, companies can tailor new product offerings, enhance service levels, and ensure customer loyalty and retention.

Future Directions

Data warehousing has greatly matured as a technology discipline; however enterprises that undertake data warehousing initiatives continue to face fresh challenges that evolve with the changing business and technology environment. Most future needs and challenges will come in the areas of active and real-time data warehousing solutions. Listed below are some future challenges:

- *Integrating Heterogeneous Data Sources:* The number of enterprise data sources is growing rapidly, with new types of sources emerging every year. Enterprises want to integrate the unstructured data generated from customer emails, chat and voice call transcripts, feedbacks, and surveys with other internal data in order to get a complete picture of their customers and integrate internal processes. Other sources for valuable data include ERP programs, operational data stores, packaged and homegrown analytic applications, and existing data marts. The process of integrating these sources into a data warehouse can be complicated and is made even more difficult when an enterprise merges with or acquires another enterprise.
- *Integrating with CRM tools:* Customer relationship management (CRM) is one of the most popular business initiatives in enterprises today. CRM helps enterprises attract new customers and develop loyalty among existing customers with the end result of increasing sales and improving profitability. Increasingly, enterprises want to use the holistic view of the customer to deliver value-added services to the customer based on her overall value to the enterprise. This would include, automatically identifying when an important life event is happening and sending out emails with necessary information and/or relevant products, gauging the mood of the customer based on recent interactions, and alerting the enterprise before it is too late to retain the customer and most important of all identifying customers who are likely to accept suggestions about upgrades of existing products/services or be interested in newer versions. The data warehouse is essential in this integration process, as it collects data from all channels and customer touch points, and presents a unified view of the customer to sales, marketing, and customer-care employees. Going forward, data warehouses will have to provide support for analytics tools that are embedded into the warehouse, analyze the various customer interactions continuously, and then use the insights to trigger *actions* that enable delivery of the above-mentioned value-added services. Clearly, this requires an active data warehouse to be tightly integrated with the CRM systems. If the enterprise has low latency for insight detection and

Comp. by: SIndumathiGalleys0000844939 Date:14/6/08 Time:00:21:27 Stage:First Proof File Path://
ppdys1108/Womat3/Production/PRODENV/0000000005/0000008302/0000000016/0000844939.3D
Proof by: QC by:

value-added service delivery then a real-time data warehouse would be required.

- *In-built data mining and analytics tools:* Users are also demanding more sophisticated business intelligence tools. For example, if a telecom customer calls to cancel his call-waiting feature, real-time analytic software can detect this and trigger a special offer of a lower price in order to retain the customer. The need is to develop a new generation of data mining algorithms that work over data warehouses that integrate heterogeneous data and have self-learning features. These new algorithms must automate data mining and make it more accessible to mainstream data warehouse users by providing explanations with results, indicating when results are not reliable and automatically adapting to changes in underlying predictive models.

Cross-references

► ETL – Extraction, ► Transformation and Loading, ► Cube Implementations, ► Query Processing in Data Warehouses, ► Data Warehousing Systems: Foundations and Architectures, ► On-Line Analytical Processing, ► Multidimensional Modeling, ► Active Database, ► Data Warehouse Interoperability

Recommended Reading

1. Kimball R. and Strethlo K. Why decision support fails and how to fit it. ACM SIGMOD Rec., 24(3):91–97, 1995.
2. Golfarelli M., Maio D., and Rizzi S. Conceptual design of data warehouses from E/R schemes. In Proceedings of 31st Hawaii International Conference on System Sciences, Vol. VII. 1998, pp. 334–343.
3. Lehner W. Modeling large scale OLAP scenarios. In Proceedings of EDBT. 1998, pp. 153–167.
4. Li C. and Wang X.S. A data model for supporting on-line analytical processing. In Proceedings of CIKM. 1996, pp. 81–88.
5. Pedersen T.B. and Jensen C.S. Multidimensional data modeling for complex data. In Proceedings of ICDE. 1999, pp. 336–345.
6. Vassiliadis P. Modeling multidimensional databases, cubes and cube operations. In Proceedings of Tenth International Conferece on Scientific and Statistical Data Management. 1998, 53–62.
7. Chaudhuri S. and Dayal U. An overview of data warehousing and OLAP technology. ACM SIGMOD Rec., 26(1):65–74, 1997.
8. Thalhammer T., Schrefl M., and Mohania M. Active data warehouses: complementing OLAP with analysis rules. Data Knowl. Eng., 39(3):241–269, 2001.
9. ACT-NET Consortium. The active database management system manifesto: a rulebase of ADBMS featueres. ACM SIGMOD Rec., 25(3), 1996.
10. Simon E. and Dittrich A. Promises and realities of active database systems. In Proceedings of VLDB. 1995, pp. 642–653.
11. Brobst S. Active data warehousing: a new breed of decision support. In Proceedings of 13th International Workshop on Data and Expert System Applications, 2002.
12. Borbst S. and Rarey J. The five stages of an active data warehouse evolution. Teradata Mag., 38–44, 2001.
13. IBM DB2 Data Warehouse Edition. http://www-306.ibm.com/software/data/db2/dwe/.
14. Rittman M. Implementing Real-Time Data Warehousing Using Oracle 10g. Dbazine.com. http://www.dbazine.com/datawarehouse/dw-articles/rittman5.
15. Kimball R. and Caserta J. The Data Warehouse ETL Toolkit: Practical Techniques for Extracting, Cleaning, Conforming, and Delivering Data. Wiley, ISBN:978-0-7645-6757-5, October 2004.
16. Linthicum R.S. Enterprise Application Integration. Addison-Wesley, ISBN:978-02016158385, November 1999.
17. Improving SOA with Goldengate TDM Technology. GoldenGate White Paper, October 2007.
18. Langseth J. Real-Time Data Warehousing: Challenges and Solutions. DSSResources.COM, 2004.
19. Paton N.W. and Diaz O. Active Database Systems. ACM Comput. Surv., 1(31):1999.