Insurance Industry Decision Support: Data Marts, OLAP and Predictive Analytics

George Bukhbinder, Michael Krumenaker, and Abraham Phillips

Abstract

Motivation. Data Warehouses and Data Marts increase the power and efficiency of an Insurance company's Business Intelligence capabilities by supporting queries, OLAP and data mining. Web-enabling of these applications makes them more user-friendly. The potential benefits greatly outweigh the costs. Data warehouse/data mart implementation streamlines information delivery for decision support and significantly simplifies development of general linear predictive models that have become more popular with actuaries and statisticians in recent years.

Method. A data mart and OLAP system was implemented for a major property and casualty insurance company.

Results. Successful implementation substantially improved the insurer's operational efficiency, providing enhanced analytical and reporting capabilities.

Conclusion. Business needs and business strategy must drive decisions about the structure and functionality of the Business Intelligence platform including the data warehouse and the data mart. The data warehouse must be well planned for the organization to realize the expected efficiencies and process improvement. However, even with a well-designed data warehouse and up-to-date software tools for data access and analysis, it is critical that the enterprise build and maintain its analytical expertise. Therefore, actuaries play a critical role in maximizing of benefits that these new tools can offer.

Keywords. Business intelligence, OLAP, data warehouse, data mart.

1. INTRODUCTION

This paper discusses certain important principles and issues in the development and operation of data warehouses from a business perspective, with a special focus on data marts, including OLAP and predictive modeling capabilities. The case study will be an actual system (the "P&C Data Mart") created for a major property and casualty insurer.

Business Intelligence, the process of gathering, storing and analyzing data, building knowledge from the analysis and taking action based on the knowledge is the single most powerful success factor in business today. In an insurance company, nobody understands data and its value better than the actuary. An actuary is in the best position to provide visionary leadership to the organization's efforts to develop analytical capability and expertise.

Consolidation within the industry, mergers, acquisitions and divestitures involving insurance and other financial services providers have created a challenging business environment. Technological advances are causing major changes in the insurance sales and distribution system. Insurance companies are recognizing the need for early detection of

changes in the environment and quick responses to those changes. In this volatile environment, competitive comparisons and data analyses need to extend beyond pricing and actuarial applications. Marketing, customer retention, sales force management, underwriting selection, pricing, claims fraud detection, loss reserving, risk management and all other aspects of the insurance business could benefit by using Business Intelligence tools.

Successful innovators in the insurance industry have been improving their Business Intelligence capabilities over the years. They have been building data warehouses and data marts. They have been using tools for on-line analytical processing (OLAP) and predictive modeling (data mining) to convert raw data into strategic advantage. They are now reaping the benefits and building on their successes.

Combining automated preparation of transaction data into account-level and more highly summarized tables for inclusion in the data mart with user-friendly means of accessing the information over the Internet or the enterprise Intranet allows the analysts to focus on analysis and research.

2. DATA WAREHOUSES, DATA MARTS, OLAP, PREDICITVE ANALYSIS

2.1 What to Expect from a Data Warehouse or Data Mart

A data warehouse is the foundation of powerful analyses. It supports business decision-making by allowing managers and analysts to examine data and perform powerful analysis easily and quickly. It facilitates measurement of the effects of various combinations of factors (geographic, demographic, rating and underwriting variables) on sales, premium, losses, loss frequency, loss severity, loss ratio, customer retention and other measures, and provides a strong platform for regression analysis and various other forms of predictive analysis.

Data warehouses, OLAP and data mining tools will not, by themselves, make a company successful. Data warehouse development must be driven by clearly understood business needs. First, the company must understand its business needs and what factors are important to its success. Then it must develop and implement ways to meet those needs. An actuary with a broad strategic vision extending beyond the confines of ratemaking and

reserves is an ideal person to lead an enterprise in developing its analytical capabilities. The ability to anticipate future needs plays a key role in the success or failure of a data warehouse.

2.2 The data warehouse

The term "data warehouse" is often used in different contexts to mean different things. In this section, we discuss three types: Corporate Data Warehouses, Data Marts, and Operational Data Stores.

Ideally, a company would like to have a "single version of the truth" in one large Corporate Data Warehouse so that all data used for reporting and analysis is extracted from it. Such a data warehouse will contain a large amount of detailed transaction-level historical data that covers multiple subject business areas brought together from multiple sources, and integrated into a convenient format for extracting information for building Data marts for individual departments and for other uses that require detailed, granular historical data. In practice, a large company may have more than one data warehouse, but not too many.

Data marts are built to address the analytical needs of individual departments. For example, while Actuarial and Underwriting areas could possibly share a data mart, Marketing may need to have a separate data mart oriented to its specific needs and the Claims Department may have to have still another data mart. Like the larger data warehouse, data marts typically contain historical data. Selected data is summarized to a level adequate to meet the intended analytical needs, for inclusion in the data mart. For example, actuaries typically do not need many items of data that might be of interest to Claims professionals. The data for the data mart may come either exclusively from a data warehouse or certain operational systems, or both.

Many experts advise against building data marts before completing an enterprise-wide data warehouse. They also prefer to have all the data for the data mart come from the data warehouse. They fear that otherwise, data cleansing efforts will be inadequate and proliferation of independent "stove pipe" data marts will result in many inconsistent "versions of the truth", resulting in indecision and frequent and expensive efforts at reconciling data sources.

Operational Data Store, unlike the data warehouse or the data mart, contains near-real time data captured from operational systems. This data is used for tactical analysis to

support on-line operations.

Data warehousing is an on-going process, rather than a "once and done" effort. As the company and the business change, the data warehouse, operational data stores and data marts need to evolve with them. New data will have to be captured, and analytical tools have to be developed and continuously improved.

2.2.1 Transactional data vs. the data warehouse

Actuaries across the industry have been using summarized, historical data bases for their analyses. However, until recently the same may not have been the case for other areas such as marketing, underwriting and claims. Even the databases used by actuaries have been limited in scope. Typically, extracting data from transactional databases has been necessary for many analyses. This approach has several drawbacks:

1. Extracting information from transactional tables can be immensely complicated.
2. Processing time would often be prohibitive, since transaction tables may contain millions or tens of millions of lengthy records.
3. Data validation and data cleansing are virtually impossible to perform while extracting data from transactional data bases for immediate analysis.

A data warehouse or data mart will contain data that has already been validated, cleansed and preprocessed at the beginning of each processing cycle, which might be weekly or monthly. This data extraction and summarization makes the data suitable for analytical queries (e.g. "What is the distribution of all of our sales by state and line of business?"). By assembling and preprocessing data we avoid having to perform the same resource-intensive steps again and again throughout the processing cycle.

2.2.2 The case for building a data warehouse

Business needs and business strategy must be the driving forces behind the building of a data warehouse. If the data warehouse does not meet compelling business needs, then don't invest the time and money in building it.

How do you make a business case for building a data warehouse? A Business Intelligence effort can be successful only in a company with visionary leadership. If the purpose of building a data warehouse is simply to automate existing processes, it may be difficult to find enough value to justify the undertaking. Automation may simply make an inefficient process

run faster. The resulting savings may be minimal. On the other hand, when innovation and business process re-design becomes the goal, data warehousing becomes part of a real Business Intelligence effort that would ultimately deliver significant competitive advantage.

The person championing a Business Intelligence project proposal will need to understand its basic precepts and believe in them. This person needs to be an excellent communicator and be willing to take calculated risks.

Trying to do something simply because "our competitors are doing it" is an approach fraught with danger. Movements such as Total Quality Management (TQM) and Re-engineering have amply demonstrated this. Every company is unique with regard to its culture, employees, customers, business processes, organizational structure and many other factors.

If budget is tight, the only option may be to start small. Build a data mart for a specific application that might, in a relatively short time, generate a high ROI from expense savings or from process redesign and efficiency improvement. Success breeds success. Once you establish your credibility, making the business case for a larger project becomes much easier.

Nonetheless, making the business case always involves identifying the business needs and the company's "points of pain" in current processes. Success in making the business case depends on how the issues are presented, the conviction and initiative demonstrated by the "champion", the credibility of the "champion" and the strategic perspectives of the "champion" and the senior executives of the company.

Some of the areas where business needs or points of pain exist may include the following:

- responding quickly to business strategies of more nimble competitors
- effectively coordinating pricing strategies and underwriting rules and activities
- performing innovative actuarial analyses (e.g. exploration of new discounts, rules-based pricing, and data mining)
- performing reliable periodic analyses of the automobile class plan, deductibles and increased limit factors, taking into account the impact of other rating variables and a changing business profile
- responding on a timely basis to regulatory agency inquiries and rate filing requirements
- understanding the causes for declining customer retention

- putting together rate filings fast enough to keep pace with competitors
- efficiently performing sophisticated analysis by state (e.g. making use of California Sequential Analysis)

A business case document should provide specific details about each need or the consequences of each problem. It should also describe how the data warehouse and related tools (such as an OLAP system and data mining tools) will enable the enterprise to address each of the issues, and how doing so could provide the enterprise with a competitive advantage.

Practically speaking, the business case will be most effective if the project's "champion" is willing to stick his or her neck out and take responsibility for demonstrating the business advantages to be gained once the warehouse and associated tools are in place.

2.3 OLAP

As with "data warehouse" and "data mart", there seems to be no single agreed-upon definition of OLAP, but a reasonably good one is "[a] category of applications and technologies for collecting, managing, processing and presenting multidimensional data for analysis and management purposes."

OLAP takes the analyst beyond pre-defined reports and allows him the freedom to delve deeper in directions suggested by the data, searching for trends and anomalies.

2.3.1 Analysis Variables (Facts) and Class Variables (Dimensions)

Analysis variables or facts are the quantities being measured. Analysis variables in the P&C Data Mart include premium earned, the number of claims, the amount of loss incurred and paid, and allocated expenses. Class variables or dimensions are variables whose effects are being measured individually or in concert with other class variables or dimensions. For auto insurance, class variables include the type of coverage (e.g. BI, PD, Comprehensive, Collision), geography (region, state and territory), coverage limits, deductibles, number of times renewed, safe driver insurance points, driver training, age or years of driving experience, and time – month, quarter, or year.

2.3.2 An OLAP Example

In an Excel spreadsheet, a user can view the effect of two class variables at the same time

in a matrix format. But modern OLAP technology allows simultaneous analysis by additional dimensions with minimal effort. In Figure 1, the measurements are earned premium and incurred loss. The dimensions selected were peril and state in rows vs. use and time in columns.

Figure 1 contains the first eight columns and all the rows in the initial screen, showing measures of analysis variables (earned premium and incurred loss) for Peril by Use. If the user wants to look at the distribution of earned premium and loss among all states for a particular type of coverage, e.g., accidental death and dismemberment, he or she simply opens up the ADD row, as in figure 2. This opening is referred to as "drilling down" through one dimension into the next. Similarly, by clicking on any "use" category, data for that use category for the various quarters could be displayed. Thus, the analyst is able to look at data from a variety of angles and continue to explore reasons for any special or abnormal results.

Without multidimensional views available on demand, analysts must draw information from various tables in the data mart or other sources and perform analysis either through user-oriented tools such as electronic spreadsheets, or performing complex database queries - a skill not every analyst has the time or desire to acquire. Multidimensional views such as figures 1 and 2 can be exported to Excel spreadsheets or other databases for independent analysis or display. Fast response and the ability of non-programming professionals to perform analysis with no programmer involvement is a clear advantage of OLAP. The advent of OLAP essentially makes standard spreadsheet reporting obsolete.

2.3.3 Multidimensional Databases (MDDBs) vs. Star schema

There are two competing OLAP methodologies based on the data structure used: multidimensional databases and star schemas.

Transactional data are typically captured in relational databases that store data efficiently in tables that are linked by primary key – foreign key configurations. Such relational database systems, designed for transactional data, were unsuitable for data analysis. This led to the development of a modified form of relational data structure called star schemas. A typical star schema is pictorially represented with a "fact" table that holds measurements (e.g. premium, losses) in the center, surrounded by tables for "dimensions" (e.g. annual mileage, use, coverage limits). Star schemas offered substantial improvement in performance in

	Use ▾	Quarter ▾							
	⊞ Bus/WL		⊞ Business		⊞ Farm		⊞ OTHER		⊞
Peril ▾ State ▾	Earned Premium	Incurred Loss	Earned Premium	Incurred Loss	Earned Premium	Incurred Loss	Earned Premium	Incurred Loss	E
⊞ ADD	$186	$	$156	$					$
⊞ APIP	$2,613	$727	$5,231	$116,740	$378	$			$
⊞ BI	$728,084	$470,816	$739,500	$34,865	$39,153	$193,646	$399,896	$342,161	$
⊞ COLL	$553,351	$273,222	$581,352	$163,662	$15,217	$42,406	$158,170	$183,222	$
⊞ COMP	$283,799	$190,524	$307,895	$61,206	$11,198	$3,869	$71,191	$14,005	$
⊞ MED	$45,461	$45,555	$37,616	$2,237	$3,157	$	$727	$	$
⊞ OBEL	$312	$	$380	$	$31	$			$
⊞ PD	$371,138	$238,722	$369,020	$93,056	$20,790	$28,757	$216,363	$121,572	$
⊞ PIP	$185,230	$48,233	$207,175	$110,220	$11,328	$3,527	$26,074	$24,289	$
⊞ PPI	$940	$	$531	$	$56	$	$22	$	$
⊞ TOWING	$4,029	$967	$6,308	$1,671	$318	$125	$453	$497	$
⊞ UIMBI	$	$							$
⊞ UMBI	$150,121	$2,470	$205,658	$1,550	$20,201	$	$55,290	$	$
⊞ UMPD	$236	$	$146	$			$	$3,973	$
Grand Total	$2,325,500	$1,271,236	$2,460,968	$585,207	$121,827	$272,330	$928,186	$689,719	$

Figure 1: Initial OLAP Screen

		Use ▾ Quarter ▾					
		⊞ Bus/WL		⊞ Business		⊞ Farm	
Peril ▾	State ▾	Earned Premium	Incurred Loss	Earned Premium	Incurred Loss	Earned Premium	Incurred Loss
⊟ ADD	AL						
	AR						
	AZ	$16	$	$20	$		
	CA			$20	$		
	DE						
	FL			$16	$		
	IA	$16	$				
	ID						
	IL	$36	$				
	IN	$20	$	$40	$		
	KY	$12	$				
	LA						
	MD	$16	$				
	ME						
	MN						
	MO						
	MS						
	MT						
	ND						
	NE						
	NM						
	NV						

Figure 2: Drill-down into State dimension

analysis situations, over traditional relational data structures. In the star schema structure, dimensional relationships are not reflected in the way data is stored physically; they need to be created from multiple tables through primary-foreign key relationships.

On the other hand, data in multidimensional databases are organized by preserving the multidimensional relationship. There are no primary key-foreign key references needed. Each fact or measure is stored as a value indexed by the values of the dimensions. The data is thus stored in a simple format and could be retrieved faster, but takes up more storage space.

In multi-dimensional OLAP, data is pre-summarized into "cubes" that are intuitively descriptive of an n-dimensional structure, the number of dimensions being equal to the number of class variables (e.g. State, age, marital status). The structure also accommodates drill-down through unrelated (age→state) or increasingly granular (region →state → territory) hierarchical structures.

Relational OLAP vendors using star schemas have made their products competitive by optimizing their performance, but for more complex OLAP queries, particularly of those involving complex calculations, the multidimensional approach may still hold an advantage.

2.3.4 Four Standards for a Successful OLAP system

Timeliness:	Requests must be processed reasonably quickly – in seconds or minutes, not hours or days
Understandability:	Reports must be self-explanatory.
Ease of use:	Analysts must be able to create reports easily, without programming
Access:	The system must be easily accessible from different locations

2.4 Predictive Analytics (Data Mining)

MDDBs are intended for exploratory analysis. More sophisticated analytical tools and models are required to derive actionable results. Unfortunately, some vendors present MDDBs as tools for data mining. They also tend to discount the need for analytical

expertise.

Of course, we could develop and set up routine processes for fraud detection and other operational uses. These processes are often designed using rules derived from extensive research and data mining. However, running the routine processes in itself is not data mining, but simply an application of the results of predictive analytics.

"Predictive analytics" is becoming the preferred term over "data mining". The convergence of technology, mathematical statistics, probability and other disciplines has resulted in highly powerful techniques for data analysis and prediction. Farmers Insurance's success with data mining, particularly the identification of the market segment of sports car owners who also own a typical family car, has become almost legendary.

In recent years, predictive modeling using general linear models (e.g. Poisson regression, logistic regression, log-linear analysis) have become immensely popular among actuaries and statisticians. Such modeling has the advantage of being more tractable and more amenable to meaningful interpretation than results from neural networks and classification tree analysis. Highly sophisticated software such as the IBM Intelligent Miner and SAS Enterprise Miner as well as many specialized software products with more limited functionality have put data mining within the reach of analysts who are not necessarily expert statisticians.

Predictive analytics can be one of the most critical uses of the data warehouse. Skillful analysis of customer data can address analytical challenges such as

- identifying new pricing variables
- finding and accounting for the overlap and interaction of underwriting and pricing variables
- assessing the impact of rate change on customer retention by market segment
- profiling and clustering current clients and prospects
- profitability models at different levels (national, state, agent), retention, cross-selling, renewal underwriting, new business acquisition
- separate models for different segments, such as tort vs. no-fault states, or "special" states (e.g., New Jersey) vs. other states
- different models by coverage (e.g. personal injury vs. property damage)

- alternatives to current insurance scoring based on variables used in credit scoring
- prospective vs. retrospective (claim vs. no-claim) models and sampling

A data warehouse can support such mission-critical business objectives effectively only if it is designed to do so. The best application software provides no benefit if the available data is inadequate or the data structure does not facilitate efficient analysis. MDDBs and star schemas[1] are not the best data structures for predictive analysis. The data structure must support easy access and data manipulation by the analytical software.

3. IMPLEMENTATION

Our discussion of implementation will consist of three sections:

1. General considerations
2. The case study: actual implementation of a Data Mart.
3. Reducing the need for re-programming when business rules change (an opportunity for future cost savings).

3.1 General considerations

We cannot overemphasize the fact that business needs are most critical in determining the contents and capabilities of the Business Intelligence system. Often such projects fail because they are built on the assumption that "if we build, they will come". Even having the right software and top management support may not provide the expected benefits unless there is buy-in from the users.

3.1.1 The actuary as visionary and essential element in implementation

The process of building the data warehouse should go hand in hand with building analytical expertise. That is what makes the visionary role of the actuary critical. Having world-class OLAP tools to "slice and dice" and "drill-down and drill-up" and access to the best data mining software for clustering and neural networks will yield little benefit if the analysts do not continually build, upgrade, refine and refresh their skills. Consultants and vendors do a disservice to their clients if they tout tools over expertise. While such tools will enhance the analysts' ability to choose analytical methods, compare results derived by

different methods and interpret results, ultimately enabling purposeful action, they are never a substitute for analytical expertise.

3.1.2 Data Warehouse requirements

Ideally, we would have one huge data warehouse that holds all the data from internal and external sources that would be shared across the enterprise. But business needs, data, reports, applications and access requirements are so diverse that the challenge of designing and building an enterprise-wide data warehouse is too formidable.

The decision about data warehouse contents may benefit from

- a review of past analyses by the enterprise
- a literature review of analyses by academics and professionals in related fields
- a review of the business environment and challenges that lie ahead
- a review of analyses that the warehouse could support

Once the content has been selected, we must identify the data sources. These could include the following internal data sources:

- existing data warehouses and data marts
- billing systems
- transactional tables
- a POS database

In addition, there are external sources:

- Choice Point
- CLUE
- Current Carrier
- ACXIOM
- Census
- RL Polk
- Regulatory information sources

- Weather data by zip/county (RMS, AIR, Guy Carpenter)
- Competitors' rate filings and territory definitions
- Other competitive information

How the data is structured within the data warehouse is determined by the intended uses of the data. The data warehouse/data mart should be carefully designed to support the desired analytical techniques, e.g., queries, OLAP, cluster analysis, regression, decision trees, and neural networks. Therefore, business users should take a very active part in the requirements gathering phase.

3.1.3 Project Management

Building a data warehouse is best accomplished through a combination of traditional Systems Life Cycle Methodology and an iterative process incorporating prototyping and double-loop learning. Although a rigid Systems Life Cycle approach helps create discipline and project stability, there will be many change requests along the way so that change management efforts could create more confusion than discipline. Double-loop learning involves thoughtful adjustment to strategies, conceptual frameworks and action plans, based on problems and issues identified. A certain amount of judiciously managed prototyping and double-loop learning will enhance the flexibility and quality of the Warehouse.

Planning the data warehouse/ data mart must include

- becoming thoroughly familiar with the data sources and the data in them
- evaluating the available software (e.g. UDB, Oracle, Sybase, SAS, RedBrick)

Business and analytical needs drive the choices of content, functionality and user-friendliness, but the data warehouse developers are responsible for prescribing the data types, table structures, methods of data extraction, data cleansing, transformation and loading, information delivery and end-user-access.

3.1.4 Extraction, Transformation and Loading (ETL)

Data for the data warehouse may come from a variety of data sources (e.g. policy

transaction files, claims files, point of sale databases, external data sources). Some data may be incomplete and some may contain errors. Data definitions may be inconsistent among different source data systems. Analytical needs may dictate different levels of granularity and summarization. The ETL process involves accessing the data, staging, cleansing and validating the data, linking data from various systems (e.g. based on account, policy or insured risk), transforming the data to fit various analytical needs (e.g. summarization and deriving measures) and loading the data to the data warehouse.

3.1.5 Tools for end-user access and analysis

The level of access should depend on the expertise and needs of the end-user. Some expert analysts may want to have access to the data in a client/server environment, enabling them to do extensive analyses. Others may prefer to have access over the corporate intranet or the Internet. Vendors often tout the capability of their architecture or products to support a variety of functionality, but much of it may be irrelevant to most users. For example, actuaries rarely if ever need access to data at the policy level with name and address information. The more complex the functionality, the more complex will be the architecture.

3.1.6 MDDB size and access time

For OLAP, an MDDB contains the value of each measure under different combinations of dimensions. The size of an MDDB increases exponentially with the number of dimensions, so it can get very big, very fast. For instance, ten dimensions with eight values each would have 8^{10} (12,058,624) combinations. At some point, an MDDB may become too big to access efficiently, since the software engine must find the answers among all of the cube's data points. Two concepts to consider:

1. Granularity. The level of granularity should be dictated by the needs of analysis. For example, policy or risk-level granularity may not be necessary for most pricing or actuarial applications.

2. Number and content of MDDBs. As with the data warehouse itself, business users prescribe dimensions and hierarchies (the latter must be specified only if the software requires these to be established before the MDDB is created), but the developers determine how best to implement these requirements. The developers must

determine how many MDDBs there should be and what class variables should be handled by each MDDB. Here, redundancy may actually help, since certain factors may be found in more than one MDDB so that those factors can be included in various combinations and hierarchies. The business users must tell the developers the number of values for each class variable, since this affects cube size and efficient cube design. For instance, the variable "state" may have 50 values while "age" (age group) may need to have only four or five bands.

3.2 Case Study: Implementation of the P&C Data Mart

The source data is obtained from a large data warehouse containing transaction-level and account-level records. These records are pre-processed and summarized on the account level and then by the various factors for use as-is for regulatory reports, queries and data mining, and for further processing into MDDB cubes.

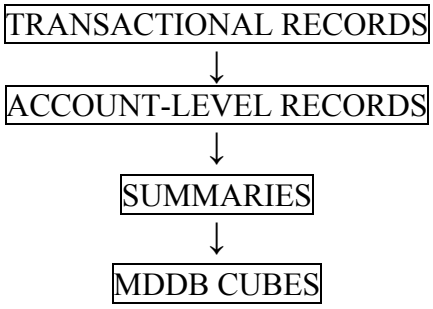
The P&C data mart and MDDBs are refreshed monthly, so all the processing described below takes place only once each month (at the beginning of the month).

3.2.1 Factors for analysis: P&C class variables

The business users selected class variables before the start of any programming. They required some changes from the way the factors are represented in the source data. For instance, source data may show the driver ages, but there are too many age values, with too little difference between consecutive values, to be useful. Little is gained in exploratory analyses, by distinguishing between consecutive or near-consecutive values. So ages were grouped into bands (e.g., 16-20, 21-24, 25-29, etc.).

3.2.2 Processing policy, premium and claims data for use in creating MDDBs

The source records included a great number of fields containing policy, customer and vehicle information. Processing the transactional records includes selecting and manipulating fields and using lookup tables to add additional more general information to each record, such as state-specific limitations or the values of certain variables based on rate class codes. End-to-end, the data goes through the following transformations, with the contents of the P&C Data Mart represented by the second, third and fourth boxes in the flow chart:



These transformations occur one time at the start of the monthly processing cycle, rather than every time information in the data mart is used to generate reports. "Summarization" refers to the aggregation of measures, such as earned premium and incurred claims, by different factors (class variables) and combinations of factors. There may be different levels of aggregation, e.g., year, quarter, and month.

The tables containing account-level records provide a base for regulatory reporting and statistical analysis (regression, data mining). The summaries and MDDBs provide a powerful platform for analyzing the effect of various factors (geography, driver age group, etc.) on premiums and claims. Whether the analyst draws upon the account-level data or summarized data depends on the inquiry. Regression and data mining will probably draw upon account-level data for use in statistical analysis. Summary data will be used for queries that do not require account-level data and for OLAP (after conversion into MDDBs).

3.2.3 Achieving efficiency in monthly processing

The entire process is summarized in Figure 3. In the source data, one record is created each month for each customer to record earned premium, and one or more additional premium records are created whenever one or more policy characteristics (e.g., persons covered, limits, deductible, perils – class variables) change or the policy is renewed. If policy characteristics do not change, then part of the current monthly record is redundant with the previous month's policy characteristic information, which would create processing and storage inefficiencies. The data mart stores this information in a form that is more efficient for both processing and storage by eliminating the aforementioned redundancy. As the monthly source data is processed into quarterly customer records, the repeating policy characteristic information ("policy history") is separated from the premium and claims

information. Each policy history record includes the start and end dates to which it applies. The starting date for that record – the date of a policy change or the beginning date of a new policy – is the transaction effective date ("TRANSEFF_DATE"). As an example, if the characteristics of a policy do not change for four months, and is repeated in each of four monthly records, we really need only one policy history record instead of four. This saves storage space, but more importantly it significantly speeds up processing monthly data into quarterly tables. The policy history dataset is in the second row of Figure 3.

Policy information is joined back to the premium and claims data such that the premium and claims amounts are each summarized for the periods during which the policy characteristics have not changed. If there are no such changes during the quarter, then three monthly records become one. The quarterly files contain account-level data not summarized to a higher level (class variables) but with reduced redundancy. They are used for data mining, other statistical analysis, and certain regulatory reports.

3.2.4 Processing monthly transaction files (premium, claims)

Now we begin to examine monthly processing in more detail. Monthly transaction files are identified in the top row of Figure 3. They contain premium and claims data for each type of coverage under each policy. They are created during the processing of source data not shown in Figure 3. (Picture the source data as being in a row above the monthly transaction files.) The monthly transaction files are then summarized into quarterly files and discarded.

The volume of transactions may be very large. For example, for auto insurance, each policy may cover several different types of coverage (BI liability, PD liability, Comprehensive, Collision, Medical, Uninsured Motorist, ADD…). Say the average number of coverage types per policy is 5. If the company has 1 million customers and wants to maintain 6 years of history by month, there would be 360 million such records (1 million customers x 5 coverages x 6 years x 12 mos/yr). This summarization means that the P&C Data Mart uses the quarter as its measure of time, rather than the month. This is usually sufficient for decision support in actuarial applications.

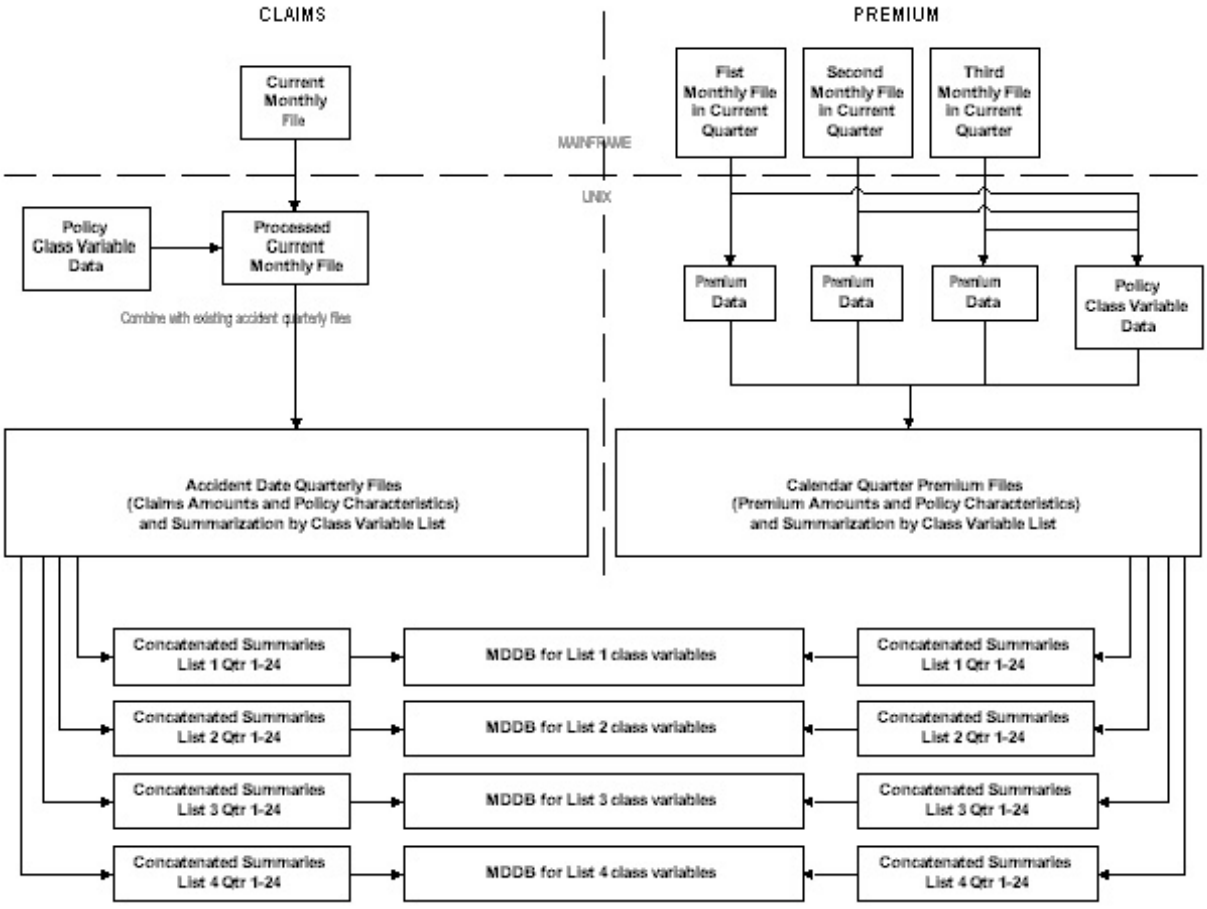


Figure 3: Overview of P&C OLAP Regular Processing

For each claim, the monthly file contains (i) customer ID and peril, (ii) cumulative incurred loss (payments + reserves - subrogation - salvage) starting from the accident date, and expenses, and (iii) deductible/limits as of the accident date. It is critical to understand that the accidents included in the monthly file will have occurred in either the current month or some prior month or even some prior quarter. To append a TRANSEFF_DATE and policy characteristics to each record in the claims file, we use the accident date as a lookup date and match to the policy history file, finding the last TRANSEFF_DATE before the accident date for that policy.

Each of the 24 quarterly claims files contains a record for each accident that occurred in that quarter. Each quarterly claims file is updated every month, with information from those records in the current monthly file regarding accidents in the period covered by that quarterly file. We do not add records to a quarterly file unless the currently monthly file

contains the first report of a particular accident in that quarter. Instead, we update records in the quarterly files by combining each record for which the monthly file has new information record with the existing record. For instance, if the June, 2004 monthly file has the record of a payment on a claim for an accident that occurred in the first quarter of 2003, we add the new payment to the claims paid total in the relevant record in the claims file for that quarter (first quarter/2003). Of course, some fields (such as payments) are additive, others (yes/no fields dealing with status) are not, but that is handled by programming.

For each policy, the monthly records for premium and exposure are combined into a new, permanent quarterly file that reflects the total written and earned premium and exposure for the quarter. TRANSEFF_DATE should already be a part of premium records, so we need not append it. For premiums, the accident quarter concept does not apply. Three current monthly files are summarized as the current calendar quarter. The prior 23 permanent quarterly files are untouched. We remove the file for the 24th prior quarter. For processing efficiency, we can store but not process the monthly premium information for the first and second month of each quarter, delaying processing until the third month.

3.2.5 Policy History

This file is represented by the box "Policy Class Variable Data" in the second row of Figure 3. It accumulates the details about the characteristics of each policy as of the transaction effective date, described above. In the case of auto insurance, the key variables for this dataset (what distinguishes one record form another) are POLICY_ID, VEHICLE_NUM and TRANSEFF_DATE. The many fields in each record in this dataset cover policy characteristics (e.g., term, effective date), driver characteristics (e.g., driver education (yes/no), age or age group, points), and vehicle characteristics (e.g., make, model year).

3.2.6 Quarterly files combine claims and premium data (respectively) with policy history

Refer to the third row in Figure 3. Policy history data is included in the records in the quarterly claims and premium files. In auto insurance, the claims and premium data would be matched to policy history by policy identification number (POLICY_ID), vehicle number on the policy (VEHICLE_NUM, say 1, 2, 3, etc., not to be confused with VIN) and the

TRANSEFF_DATE. Here again, claims and premiums are treated very differently: Premium files have many times the number of records as claims files since there may be premiums accrued for each customer each month. Only the current premium quarterly file is affected each month, but all quarterly claims files must be updated (re-created) each month. This is why we keep separate quarterly files for claims and premiums and do not combine the information until after we summarize by policy characteristics (as described below).

Claims. As described in section 3.2.3, all of the quarterly claims files are re-created every month. Fortunately, the volume of claims records is small compared to the volume of premium records, and policy history information must be appended only to the newly arrived claims records for the month. We append policy history to monthly claims data before updating the quarterly claims files.

Premiums. Even at the quarterly level, the number of records in 24 quarterly files may be huge. Fortunately, the quarterly premium files are permanent, since premium is determined for the current quarter only. Only the quarterly premium file for the latest quarter (see 3(b)) must be matched to the policy history. Most customers will have premium records for each of the three months in the quarter, and we summarize such records to aggregate monthly premium. To avoid unnecessary processing, we postpone appending policy characteristics and summarizing monthly records until the third month of the quarter. At the same time, we remove the oldest (24th prior) quarterly file from the data mart.

3.2.7 Summarization by class variable lists and concatenation of quarterly summarized files

MDDBs are manufactured from summaries described in this section. The records in each account level quarterly file are summarized by certain combinations of class variables, referred to here as class variable lists, as prescribed by the users when the data mart is being designed. This moves us from account-level records to the class variable list level.

The particular module of the underlying software system (SAS) used to create MDDBs for the P&C Data Mart required the hierarchies to be selected before creating the MDDBs. (SAS now has additional ways to create MDDBs.) For instance, if analysts will need to drill down from, say, state→age→vehicle use, then we had to summarize the records from each

claims quarterly file by that combination, along with any other combination of factors ("class variable lists") that the MDDBs must support. This takes up much of the processing time, but there are efficiency techniques available: (1) Avoiding unnecessary re-summarization of the quarterly premium datasets, and (2) using an intermediate stage of summarization. We will briefly describe these techniques.

Since the quarterly premium files do not change from quarter to quarter (except that we get rid of the oldest quarter and add the file for the quarter just completed), once we summarize one of these files we don't have to do it again unless a new class variable or hierarchy is to be added. So for premiums we only perform this time-consuming group of summarizations once per quarter, and only on one quarterly file (the new one) instead of 24. On the other hand, all 24 quarterly claims files are updated each quarter (see section 3.2.3), so we must re-summarize each of them. Fortunately, the claims quarterly files have only a fraction of the number of records of the premium quarterly files.

Summarization may be made more efficient by doing it in two stages.

Stage 1: Summarize by each of two large collections of the class variables, selected so that class variables that will form hierarchies are included in the same collection. Class variables that are found in many hierarchies may be included in both summarizations.

Stage 2: From the larger summaries, summarize by the actual hierarchies.

If the software engine does not require hierarchies to be established before creating MDDBs, and instead allows users to combine class variables "on the fly" in any hierarchical order, then summarization by specific hierarchies will not be necessary, and larger summaries, with many class variables, could be used instead. Even so, if we do not divide class variables among several MDDBs we are likely to create a single MDDB that is too large to access efficiently.

3.2.8 Concatenate 24 quarterly claims and premium summaries (respectively) for each class variable list

For each list, (i) for claims, concatenate the 24 re-created claims summary files, and (ii) for premium, start with the existing concatenated 24-quarter file, remove records for the oldest quarter, and add records for the current quarter. Result: The number of files = number of class variable lists x 2 (claims and premium). In row 4 of Figure 3, these

concatenated summaries are represented by the rectangles on the left and right side. Each such summary file now has information for all 24 quarters.

3.2.9 Merge claims and premium 24-quarter aggregate files

Merge the concatenated claims summaries with the concatenated premium summaries by each class variable list (one file per list). The common characteristics for merge are the class list variables for the corresponding claims and premium concatenated 24-quarter summaries. Now we have brought the 24-quarter claims and premium information together by class variable list. Figure 3 combines this step with the creation of MDDBs (row 4).

3.2.10 Create the MDDB cubes

Each class list becomes a set of class variables for one of the MDDBs. The analysis variables are incurred losses (and the components thereof), premium and allocated expenses. Creating the MDDBs is as simple as using a "Proc MDDB" in SAS, or VB code in the Microsoft universe, or a GUI in either SAS or Microsoft's more recent products.

3.2.11 Create the user interface

As was the case with turning summaries into MDDBs, creating a very user-friendly OLAP interface does not require reinventing the wheel. There are at least three general choices: Middleware tools (e.g., SAS IntrNet), scripts (ASP or JSP pages, using scripting languages such as Perl, VBScript or JavaScript), or canned, customizable graphical interfaces (such as those available from SAS or Microsoft).

3.3 Reducing the Need for Re-Programming When Business Rules Change

Look for opportunities to eliminate the need for future programming by building flexibility into the system when it is originally being developed. Here's an example of such flexible code included in the code for P&C OLAP System.

The value of ten class variables used in policy rating (pricing) are encoded into a short alphanumeric text string (mostly numeric) called the "rate class code". Here, "class" does not refer to the class variables in the MDDBs. The variables in question are called "rate class code variables," also called "rating dimensions". Examples of rate class code variables are age group, vehicle usage, and safe driver insurance points. Since insurance is a state-

regulated business, the business rules governing the rate class codes vary by state, and any given state's premium rating methodology involves hundreds of different rate class codes. Furthermore, the structure of rate class codes varies substantially by state. Therefore, the program code to interpret the rate class code in conjunction with the State and other factors may entail pages and pages of pure complexity. Changing this code requires a programmer and thorough testing after re-coding.

Here is another way. The user enters changes to the business rules for interpreting rate class codes through a familiar front end, such as Microsoft Excel. One of the programs used for monthly processing contains code that interprets the contents of the Excel workbook and, based on what it sees there, can change the interpretation of class codes, even adding a new variable or changing the allowable values of variables.

The programming techniques are described in an article in the proceedings of the 16th Annual Northeast SAS User's Group,[2] and the proceedings of the 29th Annual SAS User Group International.[3] Samples of the user interface are reproduced here showing a few lines of the "main" input worksheet, which prescribes the value of each variable governed by the class code in conjunction with the value of certain other variables including the state (Figure 4), a worksheet describing the characteristics of the class code variables or "rating dimensions" (Figure 5), and a separate worksheet containing the allowable values for each respective rating dimension (Figure 6). The point of identifying such opportunities is to save money in the long run, so the reader who is not a developer is encouraged to review the first of the aforementioned articles, and their developers should be referred to the second.

File Edit View Insert Format Tools Data Window Help — Type a question for help

A1 fx STATES

	A	B	C	D	E	F	G	H	I	J	K
1	STATES	RATING	Age	RATING	RATING	DIGITS					
2		DIMENSION	Group	DESCRIPTION	CATEGORY	1	2	3	4	5	6
3	CA,AK,IN,KS,KY,ME,MI,MO,NJ,NY, TX,UT	SDIP	N/A	All	All						All
4	NM	SDIP	N/A	0%	A					0,5	0
5	NM	SDIP	N/A	10%	B					0,5	1

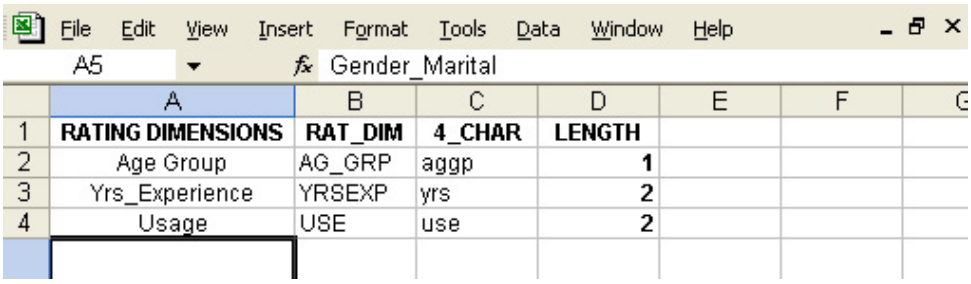
Figure 4: A Portion of the Rules Definition Sheet in Rate Class Code Interface Workbook

A more general principal statement of the recommendation made in this section 3.3 is to look for ways to separate the description of business rules from the program code, so

that users can implement changes in business rules without calling in the programmers. It's not always possible, but it needs to be considered all the time.

4. CONCLUSION

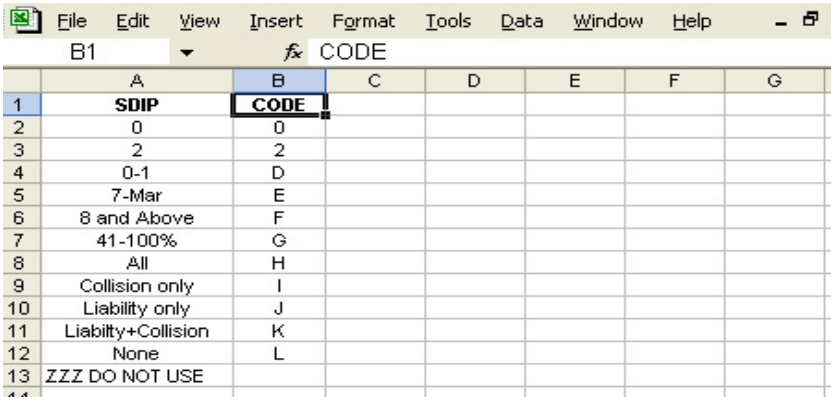
Data warehouses, data marts, OLAP and predictive analytics are essential components of a Business Intelligence system. The data warehouse enables efficient separation of historical data used for analysis from transactional databases. Business needs must drive decisions about the structure and functionality of the data warehouse or data mart. The data warehouse must be well planned for the organization to realize the expected analytical efficiencies.



	A	B	C	D	E	F
1	**RATING DIMENSIONS**	**RAT_DIM**	**4_CHAR**	**LENGTH**		
2	Age Group	AG_GRP	aggp	**1**		
3	Yrs_Experience	YRSEXP	yrs	**2**		
4	Usage	USE	use	**2**		

Figure 5: A Portion of the Rating Dimension Worksheet

OLAP is for exploratory data analysis, not data mining. Deeper analysis requires use



	A	B	C	D	E	F	G
1	**SDIP**	**CODE**					
2	0	0					
3	2	2					
4	0-1	D					
5	7-Mar	E					
6	8 and Above	F					
7	41-100%	G					
8	All	H					
9	Collision only	I					
10	Liability only	J					
11	Liabilty+Collision	K					
12	None	L					
13	ZZZ DO NOT USE						

Figure 6: Worksheet Page with Allowable Values and Formats for SDIP

of specialized tools for data mining, using advanced statistical techniques, decision trees and neural networks. Selection of software for data warehousing should take into account the needs and objectives of the overall business intelligence effort.

Even with a well-designed data warehouse and up-to-date software tools for accessing the data, the enterprise must build and nurture its analytical expertise. Actuaries are uniquely positioned to take a leadership role in maximizing the benefits of Business Intelligence tools. Furthermore, however much we may enhance our capabilities with technology, we must never lose sight of the importance of ingenuity, creativity and a solid sense of the business in analysis and decision making.

Abbreviations and notations
I/O, Input/Output
MDDB, Multi-Dimensional Database
OLAP, On-Line Analytical Processing
TRANSEFF_DATE, Transaction Effective Date (date of change in one or more policy characteristics, or the starting date of a policy.

Biographies of Authors

George Bukhbinder is the President, Palisades Research, Inc. Mr. Bukhbinder has over 20 years of experience database and data warehouse design and development, Internet-enabled information delivery for decision support, statistical modeling and data mining. He has worked extensively on the development and implementation of information systems in the insurance, financial, telecommunications and pharmaceutical industries. Mr. Bukhbinder has a Ph.D. in Statistics. He is a regular speaker at regional and national conferences.

Michael Krumenaker is Sr. Project Manager at Palisades Research, Inc. He has over six years as a full-time programmer in SAS, VBA and C++. Before that he spent seventeen years in corporate finance, including application development in spreadsheets and databases. He has degrees in business (MBA - Columbia University), law (JD - Vanderbilt, LLM - NYU), and chemistry (BS - Brooklyn), and has completed all courses for MS in Computer Science at the New Jersey Institute of Technology.

Abraham Phillips Abraham Philips is an Insurance Industry Consultant. He has an M.S. in Information Systems from Stevens Institute of Technology and a Ph. D. in Statistics from University of Windsor, Canada. He has over 25 years of experience in the Insurance industry conducting and managing data analysis, statistical research, personal lines pricing and actuarial information support including the development of data warehouses, data marts and analytical tools.

5. REFERENCES

[1] A database design that is based on a central detail fact table linked to surrounding dimension tables. Star schemas allow access to data using business terms and perspectives. University of Illinois on-line decision support glossary, http://www.ds.uillinois.edu/glossary.asp.

[2] M. Krumenaker and J. Bhattacharya, User Implementation and Revision of Business Rules Without Hard Coding: Macro-Generated SAS Code, Proceedings of the 16th Annual Northeast SAS User Group Conference, paper AD003 (2003).

[3] G. Bukhbinder and M. Krumenaker, Developing Data Marts an Web-Enabled OLAP for Actuarial and Underwriting Analysis, Proceedings of the 29th Annual SAS Users Group International, paper 111-29 (2004).