The Actuarial Process from a Data Management Point of View

Aleksey S. Popelyukhin, Ph.D.

Preface

As the information revolution advances, more information is being processed at even greater speeds and with even greater efficiency than was dreamed of only a few decades ago. Bold new approaches to information handling are being developed everyday that enable business users to arrange seemingly unrelated oddments of information into usable and cohesive databases. These databases can then be optimized and utilized across many business departments.

Advances in database technology will prove to be invaluable for the automation and enhancement of actuarial processes as a whole. These advances offer solutions to help actuaries meet the increased demands placed on them. We observe, however, that because of the varied demands actuaries pose during consecutive stages of the actuarial process, no **single** database technology may suffice. Rather an amalgamation of several of the latest data management techniques may provide the solution. We believe that the **integration** of a few major technologies, most notably Data Warehousing (DW) with OLAP (On Line Analytical Processing) and Object-Oriented Databases (OOD) with OML (Object Modification Language), may provide not only a satisfactory automation solution, but also a viable platform for future advances in technology.

Following is a discussion of these recent developments, and how they apply to the actuarial processes in the property / casualty industry. The elements to this discussion include:

- An analysis of the steps involved in the actuarial process from a data management point of view and the requirements imposed by this process on the ideal database solution
- An examination of existing database technologies in order to find the one(s) which fit better.
- A description of the "ideal" actuarial system, which can utilize technologies existing today to satisfy all the requirements discussed in the first section

A Data Manager's Problem:

Contradictory Requirements of the Actuarial Process

The actuarial process, like many analytical processes, consists of three stages: input, calculate, and report. This process presents a perplexing problem: three main stages, equally important to the ultimate goal, yet making often disparate demands upon a database.

Stage One: Input (Gathering Data and Building Objects)

Actuarial data such as losses paid, case reserves, allocated loss expenses, premiums, claim counts, etc., usually come from different sources of a transactional nature. This stage would benefit most from technologies which enable actuaries to

- Reach legacy systems,
- Extract, clean and relate data, and
- Possibly store extracted data in some organized fashion.

Thus actuaries need a technology that would enforce data cleanup, the matching of codes from different sources and provide storage optimized for future aggregations. Many insurance companies have this gathering process in place: they accumulate data into well organized tables which they aren't required to rebuild from scratch every time, but rather append regularly (quarterly, monthly, etc.). Naturally, companies don't wish to give this process up. In this paradigm all existing libraries of programs written in COBOL, PL/1, APL, and SAS programs (which read transactional data from tapes, disks, or cartridges to create datasets on mainframes or files on personal computer networks or hard drives) can be adjusted to serve as conversion tools for this type of storage technology.

From a data management point of view, most actuarial objects are results of aggregation either by time (for instance monthly data to annual or quarterly aggregations) or other dimensions (across policies to product lines, across claims to statutory lines, or across states to countrywide, etc.). On top of that aggregation, actuarial loss development triangles are essentially cross-tabulations; indeed,

- A paid loss development triangle is an aggregated cross-tabulation of payments summarized by a Loss Period (Accident, Report, or Policy-based) dimension by a Valuation Date (accounting date) dimension.

  ⇒ A cross tabulation of data by Loss Period by Valuation Date produces a right justified triangle (as in Schedule P of the statutory annual statement blank); to illustrate,

Loss Period	Valuation Date	Data
1993	1993	1225000
	1994	20000
	1995	37500
	1996	50000
1994	1994	90000
	1995	110000
	1996	800000
1995	1995	825000
	1996	1800000
1996	1996	3300000

Cross-Tabulation →

	Valuation Date			
Loss Period	1993	1994	1995	1996
1993	1225000	20000	37500	50000
1994		90000	110000	800000
1995			825000	1800000
1996				3300000

  ⇒ for a left justified triangle, the data are cross tabulated with Loss Period by Age (equal to the difference between the Valuation Date and the beginning of the Loss Period); to illustrate:

Loss Period	Age	Data
1993	12	1225000
	24	20000
	36	37500
	48	50000
1994	12	90000
	24	110000
	36	800000
1995	12	825000
	24	1800000
1996	12	3300000

Cross-Tabulation →

	Age			
Loss Period	12	24	36	48
1993	1225000	20000	37500	50000
1994	90000	110000	800000	
1995	825000	1800000		
1996	3300000			

Therefore, this task requires adequate tools optimized for aggregations and cross-tabulations. But

collections of data gathered, cleaned, and pre-aggregated into triangles or other objects are not yet ready to support reporting and decision-making. The data needs to be processed by actuaries' sophisticated methods or algorithms.

Stage Two: Calculate (Actuarial Analysis)

No two companies process actuarial data the same way. Yet we can still observe certain commonalties in the implementations of the generic process.

- First, it is usually high volume, due to the data objects being organized into many segments, to improve homogeneity of each set of data analyzed. These segmentations typically reflect a company's profile of coverages, geographic, and customer-based market definitions.

- Second, the process involves a limited number of algorithms, in the sense that the many data objects will be processed through just a few algorithms. (Some algorithms are designed to perform diagnostic testing on the data to measure trends or provide insight as to which estimation method(s) may be appropriate or not. Other algorithms process the data in order to determine numerical outcomes.)

- Third, these algorithms have a tendency to be changed from time to time. We are not concerned whether the changes are enhancements or radical departures from prior techniques; it is the **ability** to change that is imperative for the actuarial process. In other words, actuaries require an open processing system.

- And fourth, the process generally generates a finite, standard set of outcomes (estimated ultimates, IBNR, Loss Ratios, etc.), which usually have to be preserved (saved) after calculations are completed.

While many companies maintain differences, these four commonalities appear to be consistent across the industry.

The actuaries' interaction with the data and methods also sets the actuarial process apart from other, linear processes. The actuaries utilize their training and experience to interpret data and make judgments regarding certain factors. They may do any or all of the following:

- Feed their algorithms with the data, parameters and sometimes manual selections

- Calculate outcomes

- Repetitively adjust their selection, thus updating outcomes

- Store final outcomes (in either spreadsheets or printouts) either for use in further algorithms (ultimate losses for ALAE or rate adequacy analysis) or for future summarizing and reporting.

Based on the discussion above, this process demands a high volume, computationally complete, and open system, which provides the persistent storage of results and which allows for iterations and interactions.

Actuarial calculations are usually applied to matrices or vectors, which can be viewed as data objects (as opposed treating them component by component). For example, a loss development triangle is generally treated as a whole **object**; the individual elements that comprise it, on their own, do not hold much interest to an actuary. As part of the broader collection of elements that comprise the object, however, they are of great interest. The calculations that are applied to actuarial objects are generally performed on many objects, and most likely, multiple times (it would be quite extraordinary for the actuary to be satisfied by the results of the first selection).

From a data management point of view, the act of performing standard "chain-ladder" analysis is a transaction (for our purposes, an operation involving several items). As such, it is not different from crediting the interest to a savings account in a bank. For instance:

New (or modified) Interest = Average Balance Calculation Method (Account Balance, APR Rate)

Compare:

New (or modified) Ultimate = Chain-Ladder Method (Paid Triangle, Selected Link Ratios, Tail Factor)

Note, that a Paid Triangle (and for that matter the column of daily Account Balances) is treated as a whole object. In both cases some modification to an object (as a function of a few other objects and, probably, some parameters) occurs.

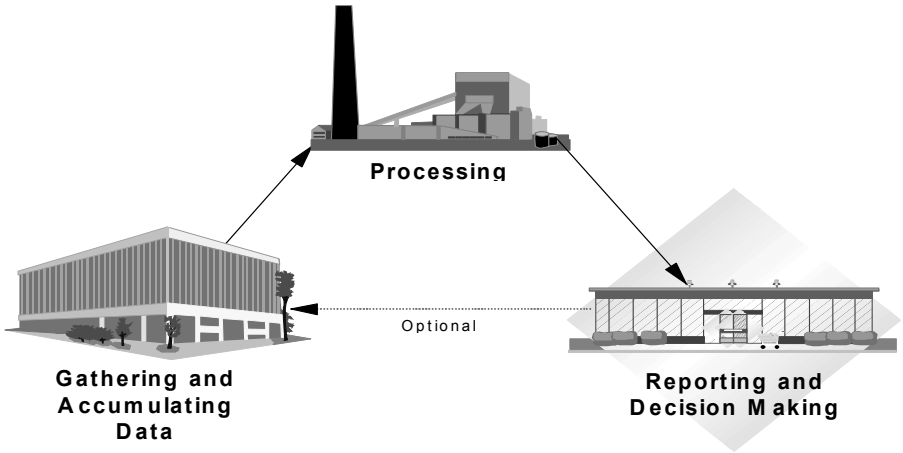
Thus, the database solution for Stage Two should be optimized for transactions on objects.

Stage Three: Reporting (Summarizing and Communicating Results)

Once the data has been gathered and calculated into results, it must, of necessity, be communicated to others in the actuarial department and to management across the company. The reporting aspect of Stage Three means summarizing and disseminating results. This allows actuarial management to review not only the final estimates, but also diagnostic summaries of the critical parameters of the process. Reporting to other departments will likely contain high level summaries of results that are relevant for corporate and line management for decision making.

The results of actuarial analysis, usually, are spread among hundreds of spreadsheet files, printouts or APL produced flat files with no mechanism to bring them together. Therefore, Stage Two needs to accumulate results in a centralized repository.

Furthermore, because the results of actuarial analysis are objects, they must be broken down into their component parts in order for them to be used for summarizations through aggregations and cross-tabulations. Thus, while breaking down these objects into their components, Stage Three should build an entity optimized for slicing, dicing, summarizing and reporting.



Technologies of Interest

We believe that, few if any, companies have achieved automation and integration of all three stages. The reason for that is apparent when you observe that the actuarial process as a whole has quite contradictory requirements.

On the one hand, during Stage One, the process requires a large storage facility optimized for aggregations and cross-tabulations, in order to generate triangles, which are essentially a cross-tab. Stage Three is similar to the first, as it also involves aggregating results, across lines and locations, for instance. These two stages would benefit greatly from the data warehousing technology.

On the other hand, the demands of Stage Two of the process are quite different. This database must efficiently store and retrieve objects (such as loss development triangles and other actuarial matrices). Its processing engine must be flexible enough to accommodate different methods and adjustments and should be optimized for sequential query execution (for processing multiple lines of business or multiple contracts) rather than aggregation or cross-tabulation. Finally, it should have an interruptible calculation process for interactive actuarial selections and judgments.

Out of all existing technologies there are two which appear to have properties that satisfy the actuarial demands outlined above: data warehousing with OLAP and object-oriented databases with OML.

Data Warehouse and OLAP

As defined by the "father of Data Warehousing" Bill Inmon (see Inmon, W.H., Using the Data Warehouse, QED, 1994), a "Data Warehouse is a subject-oriented, integrated, time-variant, and non-volatile collection of data in support of management's decision making process," where

- **Subject-oriented** means that the Data Warehouse is data-driven and organized around high level entities of the enterprise.

- **Integration** means there is consistency in naming conventions, measurement of variables, encoding structures, physical attributes of data and so forth.

- **Time-Variant** means the database accumulates historical data over time, and

- **Non-volatile** means no updates, only initial and periodic batch loading and access to the data. Every update triggers a massive rebuilding of pre-summarized sets. This restriction on the frequency of updates helps to optimize DW for aggregations.

A data warehouse is usually comprised from Operational Data Storage (ODS) which resembles a traditional data table structure, complemented with a multi-dimensional database, on which OLAP tools operate. OLAP (On Line Analytical Processing) is a technology, which allows the user to perform:

- Mathematical operations on aggregated and cross tabulated data elements (for example, IBNR = Ultimate - Reported),

- Roll-up and drill-down type of queries, and

- Pivoting, i.e. easy exchange of horizontal and vertical dimensions in two-dimensional slices.

Lotus Improv, Essbase, Holos and Excel's Pivot Tables are famous members of the OLAP category of tools.

A database technology similar to a Data Warehouse is a data mart. Roughly speaking, a Data Mart is a departmental Data Warehouse, as opposed to an enterprise-wide one.

Object-Oriented Databases and OML

An object-oriented database provides persistent storage of uniquely identified objects, complemented by a computationally complete Object Modification Language, where

- Persistent storage means that objects remain after the creation and/or modification processes have terminated.

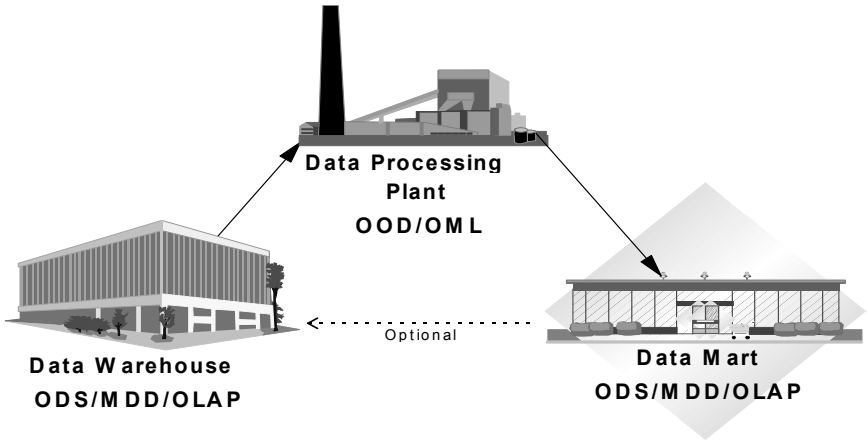
- Unique identifier reflects the object's place in the class hierarchy rather then being just the next available value as in traditional transactional databases.

- The computationally complete property of the language is emphasized because the standard structured query language (SQL) is computationally limited. Also, SQL operates on a set-by-set basis rather than a record-by-record approach, which is convenient for object processing purposes.

An OOD, with the help of an OML, provides a means for the creation and manipulation of objects and a mechanism for the storage and retrieval of those objects. Typically, an OOD is optimized for fast retrieval and updates and provides all the fundamental benefits of a transactional database to object-oriented applications: a **transactional database** for objects.

IBM DB2 Common Server, Oracle 8's cartridges, and Informix Universal Server's Data Blades are the latest commercially available additions to an OOD family.

A complete definition of OOD can be found in "The Object-Oriented Database Manifesto," by M. Atkinsos, F. Banchellon, D. DeWitt, K. Dittrich, D. Maier, S. Zdonik in the Proceedings of the First International DOOD Conference, Kyoto, Japan, 1989).



"You are what you are optimized for."

There is a clear distinction between Data Warehouses and transactional databases: namely the type of optimization they utilize. The contrasts between the data stored in a transactional database and a data warehouse are:

Transactional data	Warehouse data
Frequently changing	Static
Requires record-level access	Data is pre-aggregated into sets
Repetitive standard transactions and access patterns	Ad hoc queries with some periodic reporting
Event-driven: process generates data	Data-driven: data governs process
Updated in real time	Updated periodically with mass loads

An object-oriented database has all the properties of a transactional database outlined above, plus additional properties highlighted in the following summary of the differences between object-oriented databases (appropriate for Stage Two) and data warehouses (for Stages One and Three):

Object-Oriented database	Data Warehouse
Stores objects	Stores elements
Object identifier (non value based)	N/A
Computationally complete language	Not required

Data Warehouses are optimized for aggregations and cross-tabulations that are perfectly suited for Stage One of the actuarial process in which triangles and other objects are accumulated. A company may mass-produce triangles from one large Excel pivot table (an example of a multi-dimensional database (MDD)) by slicing it differently for various profiles of coverage and geography. The pivoting and aggregation properties of a data warehouse with OLAP are also invaluable for the reporting and decision making features of Stage Three.

Judging the properties of an object oriented database as discussed, we can see that this technology meets the demands of Stage Two for a high volume, computationally complete system providing persistent storage for calculated results. Issues of openness and interactivity will be discussed below.

The Ideal Actuarial System

Thanks to the recent breakthroughs in the areas of Data Warehousing, On-Line Analytical Processing, and Object-Oriented Databases, an ideal integrated actuarial system can be built today. However, no one of the technologies described above may satisfy actuaries completely; Data Warehouse is not optimized for transactions (triangle retrieval with saving back ultimates from the Loss Development Method is essentially a transaction), while in the OOD paradigm, it is not easy to summarize or cross-tabulate objects.

The nature of the ideal solution parallels the nature of the three stages in that there would be three main elements to the ideal data management system: the Core, an Input Converter, and an Output Generator.

The Core

In the core of an "ideal" actuarial system would lie an Object-Based storage/retrieval Database (OBD), which would store every actuarial object (triangle, matrix, row, column or scalar) as a single record in the database as opposed to the traditional approach, which is to store elements of the triangle as separate records. The advantages of such a database are numerous:

- **Speed**: Retrieval or update of one record in the database is always faster than the same operation on the multiple records. Speed is a significant factor even from an actuarial point of view: for companies that perform quarterly analyses of entire portfolios including multiple tests, either for diagnostics or estimation, speed is crucial.

- **Integrity**: Failure to retrieve or update a record with one element of the triangle (in the traditional database) may render the whole triangle unusable

- **Consistency**: Once stored, the triangle is not a subject for change because of adjustments to triangle creation algorithms or changes in line definitions. That makes it invaluable for auditing and similar purposes.

- **Diversity**: OBD can store objects of different shapes and sizes. An annual development triangle, for example, would occupy one record; a quarterly one would also occupy one record, as would a vector of ultimates. This approach is radically different from traditional "by-element" storage solutions.

- **Data Retention**: All objects, data and results are available as a starting point for the "next time" analysis or auditing.

- **Effective Storage Space Utilization**: Utilizing sparse matrix technology, triangles can be stored very effectively. (And unlike traditional "by-element" storage solutions, where descriptive information is repeated multiple times (once per element), in OBD descriptors are stored only once per object.)

- **Reuse**: Objects stored in an OBD may be reused in time (next reserve test) or in related actuarial applications (ultimates from reserving in pricing, loss ratios from pricing in reserving, ultimate losses for ALAE estimate, etc.). This reuse (sharing) of information ensures that actuaries and analysts do not expend time and effort "reinventing the wheel."

- **Application Optimization**: Most actuarial methods treat a triangle (or vector of ultimates or loss development factors) as whole indivisible objects.

The selection of a storage/retrieval system can not be considered separately from the applications. To meet the requirements of Stage Two of the actuarial process, the system has to be optimized for object manipulations and high-volume sequential query processing. In order to describe actuarial modifications of the triangles or objects of other shapes, the system needs an Object Modification Language. We assert that the language must have the following attributes:

- **Sophistication**, because some actuarial methods require complex calculations;

- **Flexibility**, because actuaries seem to perpetually tweak their methods;

- **Interactivity**, because actuaries need to get instant feedback for different assumptions, and

- **Familiarity**, because there are already too many languages to learn: APL, SAS, Visual Basic, PL/1, etc.

Actuarial algorithms expressed in this language should be size-invariant and the system should provide a mechanism for accepting objects of different sizes.

Though the demands may seem exacting, there is such a language: a spreadsheet. Spreadsheet ranges may be designated for communications with the database (accepting objects stored in an OBD and storing new, or updating existing, objects back to the OBD from spreadsheet ranges). The ideal system would provide a means to designate such ranges for exchange with the main storage facility and a re-sizing utility for adapting algorithms expressed in the spreadsheet to differently sized objects. Spreadsheet functions that are expandable through add-ins would provide a syntax for an OML. In addition, a spreadsheet's ability to interactively accept user input serves the actuarial selection and judgment process perfectly. The spreadsheet environment's printing, formatting and charting facilities only adds value to an already near-perfect match.

Note, that a spreadsheet in this scenario is treated only as a language (algorithms depository) and NOT as a file-based storage solution (as it is treated now in many insurance companies) which we suggest is non-effective. In the ideal actuarial system, effective storage for data as well as results is provided by the OBD.

Input Converter

For the core processing system to be a part of an integral solution, there should be a tool for generating pre-aggregated actuarial objects (mainly triangles and vectors) from traditional "by element" storage systems currently existing in every company.

That is where a data warehouse comes into play. A data warehouse, with its cleansed data and descriptive dimensions and members, provides the perfect platform for generating triangles and pre-aggregating other actuarial objects. Take into account that data warehousing technology is optimized for cross-tabulation and aggregation and that makes it even better suited for the task.

One way to take advantage of a data warehouse's multi-dimensional database is to use it as a triangle generator: slices of a properly organized multi-dimensional database will be exactly the triangles for different lines or other segments of the book of business. Therefore, the availability of an automated routine for triangle generation is the foundation for the ideal actuarial system. The good news is that such a tool can easily be built.

In the OBD, data objects are organized by a simple structure, most likely using the same descriptors that were used in the original data warehouse. The structure, if carefully designed, should meet the needs of most companies, yet be expandable if necessary.

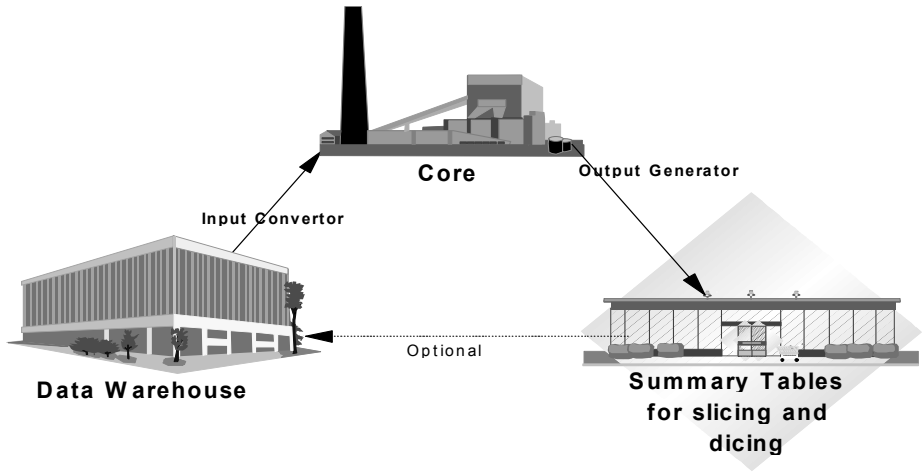
The only question that remains is how to retrieve the objects that were created during the actuarial process and stored in the OBD (for example, payout patterns or estimated IBNR)? Stated another way, how is reporting and summarizing accomplished?

Output Generator

For summarizing and reporting (Step Three), the advantages of data warehousing and OLAP technologies will again be needed. The Output Generator's task will be to break down the actuarial objects representing results (ultimates, IBNR, loss ratios) back into individual elements and create a traditional Data Mart to support decision making. Such an output generator would complete the linkage of the OBD storage system back to the data warehousing solution.

The resulting data mart could be used by itself or provide one more source for feeding the original data warehouse. As an example, the actuarial estimates of ultimate losses by line by state by accident year could be stored back to the data warehouse for subsequent use by field office management in reviewing profitability trends. Or, the estimated IBNR by annual statement line by state could be sent along to the

financial reporting department for generating Page 14 of the statutory annual statement. Plus, the original data warehouse dimensions would not be changed by either the Input Converter, core OBD or Output Generator; therefore, such a task (feeding back results from the Data Mart to the Data Warehouse) would not be a problem.



Future Enhancements

There are a number of innovations that are emerging that may prove useful to the property / casualty actuary. One of the most interesting for the purposes we are discussing here is Data Mining (sometimes called Knowledge Discovery). The ideal data management solution we have outlined above would provide a transition platform into these upcoming advancements.

Data Mining

Data Mining is a new technology on the horizon with great actuarial potential. This area of database technology applications deals with the search for regularities not known prior to the search. It examines well organized databases (like data warehouses) for clusters of similar data (and other patterns). Used properly, data mining may prove to be invaluable for determining homogeneous data subsets, which may provide clues to actuaries regarding the creation of new sub-lines or combining a few existing lines into one.

Flexible Structure

Future technology developments (Data Mining and other) may generate one more requirement for an "ideal" actuarial system: a flexible dimension-member structure, that has the ability to introduce (or delete) new dimensions or members (for example, new sub-lines) into the system.

Conclusion

The synchronization of several technologies that we have described in this paper is possible now. Once implemented, the system empowers actuaries to increase productivity tremendously and (most importantly) boost creativity. The impact of such a system would be comparable to the impact of the introduction of computers, PCs and spreadsheets themselves.

ACKNOWLEDGEMENTS

The author would like to thank Krystine Cramer, for her assistance in editing this paper; Boris Privman, FCAS, Karl Moller, FCAS and Ginda Fisher, FCAS for sharing theirs views on the current state of actuarial process automation; and Mark Littman, FCAS, for useful discussions.