A RESERVING DATA BASE

Edward M. Ford

A RESERVING DATA BASE:
Design and Implementation

INTRODUCTION

This paper describes the development and characteristics of a computer system that produces loss development and exposure data that are used in pricing and reserving at the Progressive Insurance Companies. This system does no analysis; it only produces the summary data for analysis.

In writing this, I am assuming that the reader has some familiarity with reserving data and data processing. My goal is to describe the system development process from the actuarial perspective.

COMPANY BACKGROUND

I joined Progressive in December of 1982 and assumed responsibility for loss reserving. At Progressive, the manager of the loss reserve function is the decision maker for loss reserves (there is no reserve committee) and is held accountable for their accuracy. I hope to demonstrate that the high degree of authority of this position was a major advantage in building the system.

In 1982, Progressive wrote $247 million of net premiums. All the business was auto related: non-standard personal auto, motorcycle, motorhome, travel trailer, mobile home, non-standard commercial (light to medium weight) auto, and lenders collateral protection on auto loans. The non-standard personal auto was the dominant product with 70% of the volume. The company had a history of explosive growth (in 1972, the company wrote $33 million) when the underwriting cycle was favorable, and planned to continue that approach.

The company is structured with individuals, called product managers, responsible for results (both volume and underwriting margin) for fairly small segments of business. For example, there is an Ohio non-standard auto product manager. These individuals are evaluated quite strictly on calendar year results. Quite naturally, this structure creates pressure to set reserves for these small segments individually (i.e. using the segment's development data as the dominant input).

RESERVING SYSTEMS BACKGROUND

At the end of 1982, Progressive's reserving systems produced four types of data:

1) Accident year data, ending 12/31. It contained dollars and counts of payments and case reserves.
2) Report within accident year data, ending 12/31. This was a straightforward refinement of the accident year data.
3) Accident year data, ending 12/31, for paid allocated loss adjustment expenses.
4) Case and IBNR runoff data. For any calendar quarter-ending date, the system displayed the case and IBNR emergence for individual accident years or all accident years combined. It contained dollars and counts of payments and case reserves.

After analyzing this data, reserve levels changes were implemented by revising tables in three separate systems for case, IBNR, and ALAE reserves. For more detail about Progressive's reserving approaches, the Progressive Report on Loss Reserves is available from the company.

NEW RESERVE SYSTEM JUSTIFICATION

Progressive's reserve systems at the end of 1982 were fairly extensive, and the company's reserves had been adequate historically. Why build new systems? There was no overwhelming reason, rather a number of smaller ones that jointly argued for a new system:

User perspective:

1) The systems had been built and modified with great speed because of the company's growth. As a result, there were material inconsistencies between the four data systems. For example, counts from the accident year system and the runoff system did not match. The frequency of past modifications and the turnover of the programming staff had created computer code that was almost indecipherable. It is a sobering experience to hear for the first time a smart and experienced programmer say, "I can't understand the code so I'll run some dummy records through and see what comes out." As an actuary, I am comfortable with uncertainty, but this "random variable" was particularly disturbing.

2) Reconciling these systems' data to the data used by the accounting systems for internal and external reporting was difficult.

3) The systems were not flexible enough to accommodate

Progressive's rate of change. As stated above, the company's structure forced continual study of and change in segmentation of the data. An example would be a state's rapid growth requiring separate analysis of its data. Further, the systems could not provide data in smaller time intervals than years, and for the most part the evaluation dates were limited to calendar year-ends.

Data processing perspective:

1) The systems were "fragile": a conceptually straightforward change would often produce unintended results.

2) The systems' master files were summarized and so needed to be "rebuilt" when a change was made.

3) The systems' ran very inefficiently.

DESIGN, FEASIBILITY, APPROVAL - 5/84 through 11/84

The team was comprised of myself, the two other members of the reserving area, and four data processing people (most of whom had worked on the current systems). As I mentioned above, the Progressive structure was such that I enjoyed almost complete autonomy over the system's design from a user's view. The process was simple: the reserving area described needs and the DP folks translated those needs into systems. This describing of needs sounds straightforward, but it is quite hard because people of different disciplines view the world differently and the same words can create widely divergent images. In my view this was the most critical part of the project. We made only one significant communication error (discussed below) which happily was correctable.

This process produced a schematic of the system:

1) The four current data systems would be replaced with one.

2) This one system would contain both premium (including exposure data) and loss data.

3) The table-driven system that produced the IBNR reserve at every month-end would use the data system's premium master file.

4) The premium and loss master files would be updated, edited, and balanced monthly. These files should not be summarized to minimize file rebuilding, and to provide clearer audit trails.

5) All reports would be able to show data by month, quarter, half-year, or year; and the time periods could end at any month-end.

6) The user should have control over the aggregation level (e.g. geographic, product, coverage), report type (e.g. accident, report, policy effective data), and time intervals (point 5 above).

The approval process was characterized by a disagreement within the data processing department. The people who did the feasibility study wanted to use simple flat files with COBOL; the common approach for most of Progressive's systems. They did want to use a new operating environment for efficiency and the ability to more easily create menus for user input (point 6 in the schematic above). Other data processing people argued for a "data base" approach and to consolidate this system with the system that was used to price rating variable differentials (e.g. territory, age, driving record).

I decided on the simpler approach. The decision was not that hard because the data base proponents never answered the performance and cost concerns that arose from the operation of the rating variable system.

In marketing the system to the organization, I committed to make only the absolutely minimum changes to the current systems to minimize the development costs (in other words, I promised to use the DP resources that were allocated to maintenance for development of the new system so no additional DP staff was required). I felt that I could use the current systems without modification while the new system was being built.

PREMIUM PHASE - 11/84 through 12/85

We decided to do the premium side first because it was simpler than the loss side in that there were only four statistics: written and earned premium, and written and earned exposures. There was also less concern about the quality of the historical data and there was only one source of the historical data - a file that was produced monthly containing the previous month's premium transactions.

The first task was to precisely design the reports. This naturally fell to the reserving area. A "segment" was defined as a user specified combination of company (legal entity), state, product, coverage, and limit/deductible. For example, a segment might be $12,500/$25,000 limits bodily injury for non-standard auto written in Ohio by Progressive Mutual Insurance Company. We decided we wanted

to be able to view the data on a calendar or policy effective date basis and the periods could a month, quarter, half-year, or year (see schematic point # 5 above). We also defined a limit/deductible distribution report on a calendar basis, and a "rate revision" report which combines premiums from policies that were written using a particular set of rates (e.g. rates that were revised on August 1, 1985). The rate revision view of the data was added because the pricing people often used this to judge the results of a particular set of rates.

With these report needs, the data processing people designed the method by which the user would define a segment and select a report, and the best master file structure. The segment defining and report selection was via a series of screens that the user completed.

The master file structure required a compromise. The amount of data required summarized records rather than the originally desired detail records. The record layout is straightforward. It begins with fields containing the segment information and appropriate dates and ends with a variable number of fields containing earned premium and exposure for all past and future months. The reports are then based on straightforward sort/sums on particular fields.

In building the master file, a significant misunderstanding was uncovered. In assigning an inception date to an endorsement (e.g. add a car) for policy period reporting, data processing was planning to use the endorsement's effective date. I had assumed the original policy's effective date would be the one used. I had not been clear enough in my description of the report. This reinforced the need for continual (daily) informal contact between the reserving area and data processing. Fortunately, this was correctable and as it turned out, the only significant problem of its kind in the entire project.

The testing was straightforward since there were existing reports for balancing the calendar and rate revision period data. For the policy effective date report, the testing consisted of internal consistency checks (the difference of two diagonals from a policy period matrix should balance to the calendar period report) and reasonability checks (earnings patterns and premium booking lags).

LOSS PHASE - 1/86 through 1/87

The loss side was significantly more complex on various fronts: more report types, more statistics, more complex master file structure, and less straightforward testing.

The reserving area decided on several report types:

1) Development matrices
   i) accident date
   ii) record date
   iii) record within accident date
   iv) policy effective date
   v) rate revision date

2) Case and IBNR runoff (retrospective reserve tests)

3) Size of loss

As in the premium reports, the loss reports could show monthly, quarterly, semi-annual, or annual data.

We derived twenty-one statistics. Examples would be number closed without payment, number recorded, amount paid, amount reserved.

The structure of the master file is the key to satisfying all these needs. Progressive's claim file recording structure calls for a unique claim number for an accident. For an accident, there are coverage codes and within a coverage, there are claimant numbers. The basic loss unit is called a "feature" and can be thought of as an individual claimant's cost. Each record in the master file contained all the historical information on a particular claimant's cost.

Exhibit I shows an example of the record. The record has a fixed length section at the beginning containing segment and policy information. The remaining portion of the record is variable in length and contains the transaction history of the feature. This sample loss record shows a feature that was first recorded in January of 1987 as an open case reserve (it could have been first recorded as a payment. e.g. small "fast track" payments). The reserve stayed open until it was paid sometime in April and in August, a salvage recovery was processed. There are. of course, dollar amounts associated with these items. The file is "summarized" in the sense that it shows only month-end values for reserves and month totals for payments rather than the individual transactions.

This file was built from separate files containing reserves, loss payments, salvage and subrogation payments, allocated loss adjustment expense payments, and closed without payment transactions. This single file structure is conceptually convenient because it more closely matches the way people think about claims versus the separate transaction file structure.

The building of the master file turned up a number of data problems, which caused some edits to be added to the originating, monthly transaction systems, and reinforced my intention to re-edit the data before it was used to update the master file. In general, data was not "forced", rather used to the extent possible (e.g. a countrywide report could use data that didn't have a state code). An estimate of the number of records omitted was automatically printed on some reports. I have a strong bias against forcing: it hides rather than solves problems.

Testing was a major challenge because of the lack of prior systems and the large number of statistics and report types. This step required the most time ,by far, of any in the project. Data Processing created eight dummy master records, and produced output by hand for every report and every statistic, which the reserving area checked. When the programs were written, the dummy records were processed and the output compared to the manually created output. This process was very long and boring, but worth it as a number of mistakes were caught.

MAINTENANCE AND CHANGES SINCE 12/87

Since 12/87 new transaction types (reinsurance recoverables and recovereds) and new statistics have been added. These were significant enhancements but were all done within the original design which speaks well for the design.

The regular monthly balancing/editing routines continue to catch errors in the transaction systems and have turned out to be helpful in forcing some data quality issues.

LOSS DATA FORMAT

Exhibit 1

* An accident creates a unique claim number, e.g. 453289

* A loss against a particular coverage, e.g. bodily injury, is coded with a "line coverage" code, 1910 (for B.I.)

* A particular plaintiff within a coverage is identified with an "identification" number, e.g. 01

* So a "feature" is a unique combination of claim number, line coverage code, and feature identification code, e.g. 453289 1910 01

Sample record – one feature

Segment/Policy Info	RSF 8701	RSF 8702	RSF 8703	PDL 8704	SAL 8708

APPENDIX

This appendix is a more technical description of the system and was written by Phil Juarez.

I. Overview

The LODESTAR (LOss DEvelopment STatistical Analysis and Reporting) computer system's primary purpose is to act as a collection point for premium and loss data and to provide a facility by which this data can be used for loss reserving and pricing. Conceptually, LODESTAR is a database management system, albeit a primitive one. The "database" is updated monthly and the,"Fourth Generation" reporting mechanism is provided through CICS and batch programs.

LODESTAR was developed using COBOL and Command Level CICS (at the time version 1.5) in an IBM mainframe (MVS) environment. The system consists of over 50 in-house developed programs (30 batch and 20 CICS). The system is logically divided into two sub-systems, Premiums and Losses. Each sub-system has a monthly file update and an on-demand reporting facility.

The cornerstone of LODESTAR is the file structure. All the datasets are sequential tape and/or disk files. The master files have a variable (RECFM=VB) record format. This allows both storage economy and dynamic record growth. The update process simply appends the current month's transactions to the end of the record. The result is a record which contains the complete history of a claimant.

II. Monthly Master File Updates

Balancing to the Accounting Systems

Each LODESTAR sub-system (Premiums and Losses) requires that the transactions generated during the month be added to the appropriate master file. These transactions are created in the source systems (i.e., Premiums, Claims, etc.) and passed to a number of downstream systems including LODESTAR and the General Ledger (MSA) system. Because of problems reconciling the pre-LODESTAR loss reserving systems to our accounting systems, it was a design requirement that this reconciliation occur prior to updating the LODESTAR master files. This is accomplished by reading the actual transaction file and the General Ledger "posting" file and insuring that the two are equal. This automatic balancing occurs prior to the file updating and if an out of balance condition occurs, the program is cancelled and a programmer is called to correct the problem. There have been numerous benefits of the balancing process but the most significant is immeasurable: the company-wide confidence in LODESTAR's

data. From this point the Premium and the Loss updates are significantly different and will be discussed separately.

Premium Master File Update

The Premium system at Progressive is a transaction driven system with no master file. Therefore the data LODESTAR processes the actual New (or renewal) Business, Cancel and Endorsement records. Upon completion of the balancing the New Business records are separated from the Cancels and Endorsements and used to update a "Policy Inception Date" master file.

The Cancel and Endorsement records are then matched to this file to attach a policy inception date to these records. All the records are then merged and summarized to the company, state, product code, coverage code, policy effective date and policy expiration date level. We view this summarization as one level above the policy detail. This summarized transaction file is used to update the LODESTAR Premium Master file.

The summarized transaction records are converted to LODESTAR master file format and merged with the current version of the file. The LODESTAR master record consists of a "fixed" area and a "variable" area. The fixed area consists of the key noted above plus policy inception date, rate revision date, written premium and exposures. The variable area contains the monthly earned premium and exposures.

As noted above, another design requirement was the calculation of the IBNR reserve (by applying factors to the past earned premiums) directly from the LODESTAR premium file. This file structure easily associates the earned premium to the time period for which it was earned and permits direct calculation of the IBNR reserve.

Loss Master File Update

While the LODESTAR premium file gets all its data from one source, the LODESTAR loss file get its data from many sources. Loss data includes reserve, paids, salvage, subrogation, ALAE paid, reinsurance recovered and recoverable. All these sources are balanced against their "posting" files and then converted to LODESTAR master file format. The key for the loss file is company, state, accident year, claim number, coverage code, and claimant number. This basic loss unit is called a "feature". Other information found in the fixed area of the record include the policy number, policy effective and expiration dates, and rate revision date. The variable portion of the record contains a complete loss history of any given claimant.

After the current month's information has been appended to the appropriate record the record is programmatically examined to determine if the feature has reopened or if the transaction is a continuation of the current feature. This dynamic feature analysis also determines if a feature has changed status (e.g. closed without payment) in other ways since the last update.

III. On-Demand Reporting Facility

The ability to select the "segment" (a selected group of companies, states, products, coverages, and limit profiles) was an important design requirement. Fixed or pre-defined aggregation levels were not an acceptable alternative. Another important design requirement was the ability to select statistics (e.g. number of paid features, dollars of paid losses). The solution was a design that incorporated CICS to both edit the selection criteria that the user computed via pre-defined screens and to submit batch jobs to the JES internal reader.

Upon completion of the screens above, a batch job is submitted to generate the desired report(s) at the level of detail requested.

IV. Batch Report Generation

The on-line segment, statistic, and time period selection facility permits the user to select what data he or she wishes to analyze in up to ten different report formats. The request is then translated to a batch job which actually completes the request. The batch job has two fundamental components: extraction of the requested data and reporting of that data.

Extraction is accomplished by reading the master file and comparing each record to each of the up to twenty separate report requests from the on-line. If the record matches one or more of the requests it is flagged to indicate which segment it matches and is passed on to the reporting section. This method has two advantages: the very large master files are only read once and the resulting extract file has no redundant records.

The extract file is input to any or all of the report programs. A report program will perform an internal sort of the extract file, sorting only those records which are flagged for that report. During the sorting process the aggregation levels defined in the on-line are established in the sort-key. Therefore, upon sort completion, a simple control break logic can be utilized and the report can be generated.

The process defined in the preceding paragraph is repeated for each segment requested from the on-line. The extract file is therefore passed for each of these segments. Depending on the request, this can be significant.

V. Report Output

Finally, the user has several choices in output type. The user selects one or a combination of several media when the pre-defined screens (Section III above) are being completed.

The choices are essentially paper, a file for downloading to a PC (for importing into spreadsheet software), or a file that becomes input to analysis routines programmed on the mainframe.