SOCIETY OF ACTUARIES
MODELING SECTION

The Modeling Platform

ISSUE 10 • NOVEMBER 2019

The Importance of Centralization of Actuarial Modeling Functions, Part 1

By Bryon Robidoux

Page 8

The Modeling Platform

Issue 10 • November 2019

Published biannually by the Modeling Section of the Society of Actuaries.

475 N. Martingale Road, Suite 600
Schaumburg, Ill 60173-2226
Phone: 847.706.3500 Fax: 847.706.3599
www.soa.org

Publication Schedule
Publication Month: April 2020
Articles Due: Jan. 24, 2020

The digital edition of this newsletter can be found at https://www.soa.org/sections/modeling/modeling-newsletter/.

Chairperson's Corner

By Brenna Gardino

Time sure does fly when we are busy, and the past five years have been no exception. It was recently pointed out to me that the Modeling Section has its fifth anniversary this year. Wow, five years! Our section has certainly been productive and successful in that time period.

OUR HISTORY

On May 23, 2014, a letter was submitted to the Society of Actuaries (SOA) Board of Directors from the Organizing Committee for a Modeling Section to approve the formation of a new section, the Modeling Section. The proposed mission statement of the section was:

> To support the basic and continuing educational, research, networking and other specialized needs of its members that relate to any aspect of the creation, governance and use of models.

The committee included:

- Trevor Howes, chair
- Bruce Rosner, vice chair
- William Beatty
- Teresa Branstetter
- Mary Pat Campbell
- Tim Cardinal
- Shane Leib
- Jason Morton
- Zohair Motiwalla
- Phil Schechter
- Eric Schwartz
- Suzanne Voorhees

> "… a modeling section will have a common interest that crosses national boundaries and all sizes and types of employers …"

The letter noted, "In recent years there has been a widespread and significant growth in the power, complexity and usage of models for all purposes." Five years later, this feels as true, or even more so, with regulatory changes such as Generally Accepted Accounting Principles (GAAP) long-duration targeted improvements (LDTI). The letter also stated, "Because modeling principles are independent of product and application, a modeling section will have a common interest that crosses national boundaries and all sizes and types of employers, including independent actuaries and academics." This aspect of our section has also proved to be true, and we have fostered an environment of collaboration across disciplines with core topics applicable to a wide range of individuals.

OUR GROWTH

Once the board approved the formation of the section in 2014, the committee had 12 months to obtain 200 paying members. Growth of the Modeling Section came quickly.

- July 2014: 89 members
- August 2014: 345 members
- December 2014: 387 members
- July 2015: 1,239 members
- December 2015: 1,356 members
- December 2018: 1,579 members

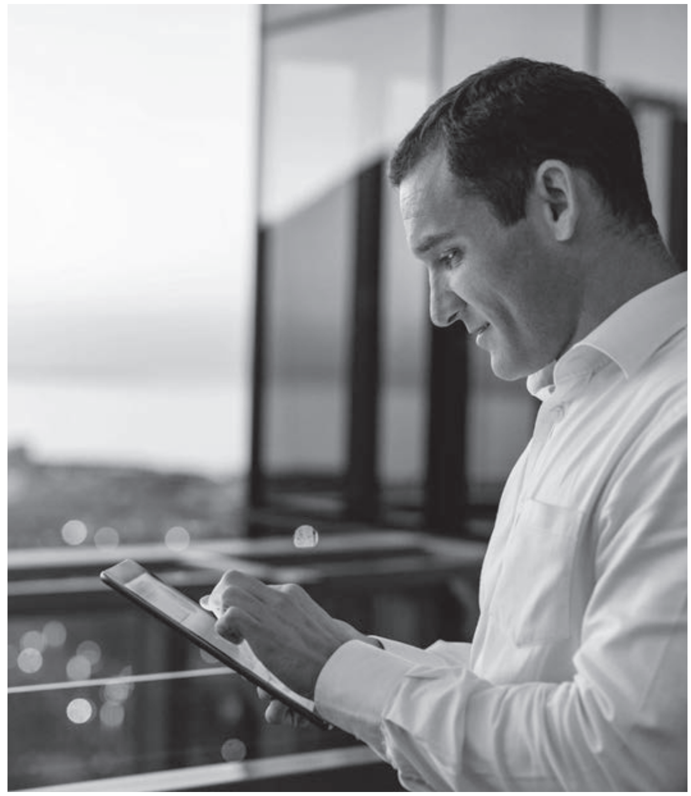
Over the five-year period, the Modeling Section grew from the smallest section of the 20 SOA professional sections to the eighth largest. I anticipate continued growth as we continue to offer strong educational content and networking opportunities.

This past year has been a terrific one, and I am grateful to have been involved. Modeling Section highlights include: the offering of economic scenario generator and GAAP LDTI webcasts, the addition of a monthly modeling hot topic discussion, strong SOA meeting and newsletter content development, and the recent Modeling Excellence workshop. These efforts have been successful because of active and passionate section volunteers. In particular, I want to thank Daphne Kwan, the Modeling Section's incoming chairperson, for her support and hard work over the past year. Please reach out to the section leadership to offer ideas, get even more involved, and to help make the next five years as productive and successful. ■

Brenna Gardino, FSA, MAAA, is a senior consulting actuary at Actuarial Resources Corporation in Overland Park, Kansas. She can be reached at *Brenna.Gardino@arcval.com.*

Letter From the Editors

By Mary Pat Campbell and Jennifer Wang



Welcome to the fall issue of *The Modeling Platform*. We have quite the variety of articles for you, including some digital-only content.

DON'T MISS THE DIGITAL COPY

We're experimenting in this issue, trying to take advantage of what may work better in a digital format rather than printed on paper.

Our regular feature describing modeling-related SOA meeting sessions is in only the digital edition, as is the second part of the article "Confessions of an Efficiency Junkie."

There are a few ways you can find the digital copy, but the easiest way is to go to the Modeling Section area of the SOA website and navigate to the newsletter. The link to the newsletter page is *https://www.soa.org/sections/modeling/modeling-newsletter/*.

The first issue we had in digital format was back in the spring; the same content was in both the print and digital editions. We are now trying some digital-exclusive content, which helps us share more information in a more efficient way.

CONTINUE THE CONVERSATION

While we're wrapping up the editing of this newsletter, we're also on the prowl for more authors and more articles for our first 2020 issue. In this issue, Bryon Robidoux shows one method of producing submissions: responding to prior articles. "The Importance of Centralization of Actuarial Modeling Functions, Part 1" continues the conversation started in the spring 2019 issue of two articles on centralization of the modeling function within an insurer. As the title indicates, there are further parts coming, so look for those in 2020.

Other inspirations can be changing standards and regulations, as we see in the articles "Play Ball! Modernizing for LDTI" and "Insights Into Life Principle-Based Reserves Implementation and Modeling Practices." The first looks at upcoming accounting changes for U.S. Generally Accepted Accounting Principles (GAAP), and the second looks at a change already in progress.

We also see articles coming from experience, with principles and practices the authors have found helpful. "Confessions of an Efficiency Junkie" looks at what is really involved in trying to make processes efficient … and what that even means. Again, don't miss part 2 online. "Writing Effective Model Documentation" looks at not only how to document your models but also what goals one is trying to achieve with this documentation.

Other articles have some high-level principles to keep in mind in modeling. "Key Principles of Actuarial Model Governance" summarizes key goals developed among many actuarial groups. "Economic Scenario Generators, Part I: Motivation for Stochastic Modeling" helps lay out why one would want to model stochastically. Note that's also the first of a multipart series as well—keep an eye out for more articles there.

Finally, a tool that we can use: "An R Package for Experience Studies." The R package described in this article can be seen (and forked) at GitHub. As we write this letter, there are various notes on the code—in GitHub, one can propose fixes and additions to the package author. Please share with the actuarial modeling community (non-proprietary, non-confidential) projects you're working on. With increasing use of R and Python in the world of data science in general, there may be some public repositories one can share. It need not be your own code, even. There is a growing number of packages and code out there, so "curating" a list of what you find helpful will help the actuarial community.

SHARE THE LOVE

If you enjoy or appreciate an article, please contact the authors and let them know.

Authors have their articles published, and while there is a warm glow of seeing one's name in print, one wonders if anybody is reading the articles, and if they do, what they think about it.

Alas, our digital version does not have a "like" feature (yet), so in the meantime, why not drop a quick note to the authors? With contact emails listed at the end of each article, it's a snap. Go do it. Heck, just copy/paste this line: "I just read your article in the fall issue of *The Modeling Platform*—I really enjoyed it!"

If you didn't like an article, or disagreed with what the authors had to say, why not write your own article with your perspective and send it to us, your friendly editors? Keep the ball rolling. ■

Mary Pat Campbell, FSA, MAAA, PRM, is vice president, insurance research, at Conning in Hartford, Connecticut. She can be reached at *marypat.campbell@gmail.com*.

Jennifer Wang, FSA, CERA, MAAA, is a consulting actuary at Milliman. She can be reached at *jennifer.wang@milliman.com*.

The Importance of Centralization of Actuarial Modeling Functions, Part 1: Focus on Modularization and Reuse

By Bryon Robidoux

This article is a response to the April 2019 issue of *The Modeling Platform* that contained discussions on the pros and cons of centralizing and decentralization the modeling departments.[1] This article will look at these activities from a software engineering perspective. It will give clear insights as to why large corporations, such as Prudential and New York Life, would find it critical to centralize their modeling departments and explain how to get the most out of centralization.

Why would Prudential and New York Life want to spend their time, money and effort to centralize their modeling departments? As an insurance organization grows, especially to a very large size, it becomes more likely that different products will be splintered into silos to better focus on the product lines. This makes sense from a product management standpoint, but it starts to create problems with modeling.

To see why, let's create a fictitious insurance corporation called ZZZ. When ZZZ opened its doors 10 years ago, it offered fixed indexed annuity (FIA), variable annuity (VA) and universal life (UL) products and put them in independent branches of the organization. Today, they decide to offer indexed universal life (IUL) and variable universal life (VUL) products and have the UL team model them. The options and option budget calculations are going to be similar between the FIA and UL plans. Similarly, the VUL is going to offer the same set of mutual funds, especially volatility-controlled funds, as the VA line so their account-value behavior is similar. What options are available to ZZZ as they try to model their new product offerings?

There are three ways to move forward in this situation:

1. Start from scratch
2. Copy existing code
3. Build reusable libraries that can be shared

OPTION 1: STARTING FROM SCRATCH

Starting from scratch is hardly ever a viable option, even if there is a strong desire to do so. This can be a very expensive and risky endeavor because, no matter the good intentions of starting with a clean slate, there is a good chance that reality will set in and the new model will start having all the same blemishes as all the other models in the organization.

In behavioral economics, it has been shown time and time again that there is a bias to underestimate the time and effort needed to accomplish a project even with experienced professionals. [2]

OPTION 2: COPYING EXISTING CODE

Copying actually has two paths it can follow. The team can either copy and periodically synchronize with the original, or they can copy and modify by throwing out calculations or adding new calculations.

Synchronizing is very challenging and no small task due to continual coordination and reconciliation. Copying and then modifying is the most likely option due to stakeholders wanting to work independently to manage their own priorities. Actuaries are motivated to copy because it appears to be the cheapest and easiest to implement. Copying and decentralization are really one and the same activity.

Copying appears to be cheap because of the divide-and-conquer fallacy. This is the idea that a model can be copied to better divide and conquer the workload so that deliverables can be parallelized, finished independently and faster. Copying actually increases the workload because each copy takes on a life of its own. The models will have to be developed, tested, run, audited, controlled and managed separately. The model will start diverging due to inconsistencies, at worst, that should not exist or, at best, are annoying. All the differences manifest themselves with various models giving different results for what is supposed to be similar or identical behavior.

With the insurance companies offering ever more complex products and dealing with ever more complex regulation, senior managers need to eliminate as much noise as possible. They should not have to be thinking which model produced the results and trying to mentally juggle the differences.

Even if the calculations remain identical, the more the original and the copied model diverge, the more infrastructure and adaptions are required to release the model into production. If the only requirement for a model to go to production is that it spits

out accurate numbers, then the infrastructure needed to feed the model will be overlooked. This will result in manual, tedious and error-prone processes. The worst thing that can happen is an actuary creates one, two or 10 spreadsheets to bridge the gaps between model inputs or results.

Further, the design of the model is directly related to the service level that can be provided to stakeholders. The stakeholders will want to do what-if and other analysis outside of the normal production runs. This additional analysis will likely take forever, requiring an army of people, or the results will be unreliable if the manual adaptation gets out of hand.

Decentralization: An Expensive Choice

When starting from scratch or copying, there are multiple groups effectively maintaining the same functionality and solving similar problems. In the end, what had been done to reduce timelines has just increased work, headcount or both. Once a model goes to production, the process dictates the structure of the company and not the other way around.

With copying, I often relate the project to a tractor pull. It starts fast, but eventually the sled weight will bury the tractor in the mud. The fast start will lead to short-term decisions that don't scale. As the project grows, it will start being very complex and error prone. Eventually, new feature delivery will grind to a halt and stakeholders will find new sources for their results. Copying is therefore a short-term gain for a very expensive long-term loss, which makes the pros of decentralization a mirage. Just like in finance, there is no such thing as a free lunch. The piper wants his payment and his wealth derives from immediate gratification from making copies!

To achieve a smaller, better, faster and cheaper modeling operation, centralization is the correct move. To really get the most out of centralization, the goal must be to focus on modularization and data and logic reuse by using software engineering practices.

Applying Software Engineering Concepts

In actuarial modeling departments, there seems to be a mental separation between software engineering and model development. In reality, there is no difference. To demonstrate the point, here is the mapping of roles from the modeling department to the software development department:

- Actuarial modeler ⟶ developer
- Model steward ⟶ application technology lead

- Model governance ⟶ development operations
- Stakeholder sign-off ⟶ user acceptance testing
- Setting up runs and their switches ⟶ configuration management

The list is by no means exhaustive. The pain we feel in the actuarial modeling department is directly related to the difference we believe there is between the activities. The software engineering profession already has a lot of our modeling challenges solved.

While possibly provocative, I think the modeling department should report to the chief technology officer rather than the chief actuary. The modeling department is more an extension of the modeling platform than the insurance organization itself. Actuarial modelers are just customizing the platform so the business can run the calculations it needs.

> To really get the most out of centralization, the goal must be to focus on modularization and data and logic reuse by using software engineering practices.

A sound modeling department needs to be a wide spectrum of technology and actuarial skills all working together. New regulations have sophisticated requirements. Senior managers need to have the flexibility to look at numbers and do whatever analysis is required to make better and faster decisions. This can't be done with cumbersome and clunky models with a million manual processes.

Software engineering is the art of abstracting sets of related concepts so they can be dealt with in a uniform manner. Good software engineering involves the following key aspects:

- One should focus on modularization.[3]
- Each of the units should be tested to ensure they work properly before being merged into the main production branch.
- Units should be simple reusable components.
- Each component is divided into abstractions that semantically map to the problem at hand.
- One should develop to pattern, not to the specific problem.

A project backed with good software engineering practices will start slow like a large rocket and then accelerate once a critical amount of functionality is built. The idea to remember is that nothing is ever new. It is usually just a slight extension on what already exists. With a little patience, focusing on work product reuse will pay large dividends and allow the modeling department to do more with less and easily adapt to new changes.

Modern project management, such as agile development, is based on the axiom that good engineering practices are modularized. This is why there are one- or two-week sprints to build small units, get buy in, make necessary adjustments and move on to the next small units of work. This very iterative process leads to a much better product with faster feature delivery.

OPTION 3: BUILDING REUSABLE LIBRARIES

Now back to the last option for company ZZZ, which is to build reusable libraries for common calculations. This means that the group deemed to be the subject matter expert builds a library and shares it with the rest of the corporation so that everyone benefits from the expertise. Rather than organizing the insurance company in terms of product lines (which can have many redundancies), the company can be organized in terms of common services. This service-oriented corporate structure would strive to only do a task once, have only one source of data and make code easily extensible to avoid redundant logic. For example, there would be a team responsible for providing the option and option budget calculations for both FIA and IUL products. This promotes consistency and greatly speeds up the rate at which enhancements can be added to models.

Code modularity and work-product reuse are not easy to implement. They require coordinating with other groups and living within their response times. There is also an entire discipline of software engineering above and beyond actuarial science to learn. Our modeling platforms try to shield us from software engineering, but this effort is futile. Trying to shield modelers from software engineering is equivalent to playing the whack-a-mole game. By hitting one mole, it will cause another to pop up somewhere else.

If software engineering, modularization and work product reuse are so important, then why are they not common practice in the actuarial modeling department? By the design of our modeling platforms, actuaries are strongly encouraged or mandated to put all their work products directly into the model. Once in the model, it is locked away from other projects that might need that same logic. This is the monolithic-system problem.

The monolithic-system problem forces modelers to copy logic, which causes issues mentioned earlier in this article. It also creates challenges with unit testing and project management because it makes unitizing the work very difficult. (An example of a unit of work might be a formula table in Axis or an extended formula in Prophet.)

In our models, many units are merged together, there are tons of switches that have to be set correctly, and many components have to be put into place in order to get the model to run. These components have to be set up just perfectly to make sure that all the execution paths are exercised during testing. This requires actuaries to put changes into the model and then sort out all the issues. This forces the testing team to run the entire model to find problems. Depending on the size or number of run(s), this could be a multiple day turnaround to analyze changes, determine the sources, explain to the developers any issues, have developers fix and/or dispute the perceived problems, and get the model back for the next round of tests. This leads to project management issues, especially in agile project management. Depending on the complexity, it is difficult to build the units, assemble then together and test within a typical one- or two-week sprint. The sprint can be lengthened, but this cuts down tremendously on the agility of the project.

This is backward to what should be done. With proper unit testing, the actuary would find all the problems *before* the model is fully built. The tests should take seconds and not days. The developers should be able to run independent of the testing team so they can get very fast turnaround. Then all that would be required is a little bit of integration testing to make sure all the units play together nicely with the model. This is much less work than the current practice.

CONCLUSION

The expense of decentralized models stems from three main problems: the perceived difference from actuarial modeling and software engineering, the divide-and-conquer fallacy and the monolithic-system issue. The act of copying the model to try to divide and conquer the workload actually creates more work. This cheap act of copying creates a massively expensive modeling department and does not scale in the long haul. Decentralization and copying are the antithesis of sound modeling and good software engineering practices.

By not accepting that actuarial modeling and software engineering are the same jobs, the actuarial profession is struggling with problems the software engineering profession have already solved. If we harness software developers' expertise and their tools, we can reduce many of the challenges we face.

Centralization of modeling is a good start, but it isn't the end game. To make the centralization really pay off, modeling departments need to go one step further and focus on modularization so that logic and work products can be reused as much as possible. This will speed up development throughput and testing. It will make it easier to audit, document and maintain the model. If done correctly, this will reduce unnecessary head count and make the models smaller, better, faster and cheaper to operate and maintain. ■

Bryon Robidoux, FSA, CERA, is actuary ALM, Reinsurance Group of America. He can be reached at *bryon.robidoux@rgare.com.*

ENDNOTES

1 Kwan, Daphne. 2019. Journey to Centralizing Modeling Function. *The Modeling Platform*, no. 9. *https://sections.soa.org/display_article.php?id=3361692&view=582866*; Kerr, Dean, Josh Chee, and Jay Boychuk. 2019. Centralizing Model Development: Is It Worth It? *The Modeling Platfor*m, no. 9. *https://sections.soa.org/display_article.php?id=3361693&view=582866.*

2 Kahneman, Daniel. 2011. *Thinking, Fast and Slow*. New York: Farrar, Straus and Giroux.

3 "Broadly speaking, modularity is the degree to which a system's components may be separated and recombined, often with the benefit of flexibility and variety in use." *Wikipedia*, s.v. "Modularity," accessed Aug. 29, 2019, *https://en.wikipedia.org/wiki/Modularity#cite_note-MWModular-1.*

Play Ball! Modernizing for LDTI

By Tim Koenig

Over the past couple of years, companies have been gearing up to comply with new long-duration targeted improvements (LDTI) guidance from the Financial Accounting Standards Board (FASB). Some waited too long to start discussing accounting policy decisions and their resulting impact on actuarial model(s), thereby increasing execution risk, and have felt like they're down in the bottom of the ninth inning. On the other hand, some started early. Having already made planning progress and now nearing an execution phase, perhaps they feel like it's a tie game in the top of the fourth, unsure of what the future innings have in store.

RAIN DELAY

In mid-July, regardless of the ballgame status, it rained. On July 17, FASB announced a deferral of the compliance deadline: "a one year deferral for larger SEC [Securities and Exchange Commission] filers and a two or three year deferral for all other entities."[1]

What does this mean?

For the companies in the bottom of the ninth, their game plan doesn't change, they just get some rest before those last few at-bats, as they now have more time for testing model capabilities and planning for the scenario of a longer-than-expected implementation. The companies in the top of the fourth, however, have an opportunity to treat this rain delay as a downpour. In this extended metaphor, it is conveniently the fourth inning, so a long rain delay results in a new game, a pivot to modernization.

> FASB announced a deferral of the [LDTI] compliance deadline … companies have an opportunity to treat this rain delay as a downpour.

What does using LDTI to modernize look like?

- Converting valuation and projection processes to a new modeling platform poised for LDTI compliance
- Building a new "one-source-of-truth" database solution
- Automating feed of data and assumptions to model
- Feeding post-processing calculation engine and automated results to LDTI disclosure templates
- Automating reporting dashboards of LDTI metrics using new technology such as Tableau or Power BI

These are very high-level bullets, and one could write entire articles on each. Rather than hitting on each of these, here are two brief pieces of advice for "managers" to consider as they strive to improve their team—although not as profound as Yogi Berra's "Baseball is 90 percent mental; the other half is physical," or "When you come to a fork in the road, take it."

ESTABLISH SIGNS AND KNOW THE STRIKE ZONE: IDENTIFY DEPENDENCIES

Smack! The ball hits the catcher's mitt for a called strike three on the outside corner. This all happened due to a string of dependencies. For instance, the pitcher depended on the catcher to signal for a pitch that exploited the batter's weaknesses, and the catcher depended on the umpire to maintain consistency in his strike zone. Furthermore, these dependencies may have been identified the hard way. Perhaps the catcher struck out looking at an outside pitch his last at-bat and noted this for the future.

When it comes to complying with LDTI, companies can't afford to strike out or forfeit home runs to realize they did not properly identify and act upon dependencies. Rather, they must identify them early, before the implementation phase of the project. Many modernization projects involve multiple workstreams. Just as the pitcher and catcher must communicate, the modeling workstream must communicate with the accounting policy workstream, the accounting policy workstream with the data workstream, and so on. Consider this scenario. The accounting policy workstream is deciding to set the discount rate using either a forward curve or average rate. Based on impact estimations and other discussions, they decide to go with forward rate. A few months later, when the modeling team goes to incorporate this decision, they discover the model has limitations and can only accept an average rate. Other project tasks were dependent on this task, and one missed communication point ultimately resulted in a project delay.

Utilizing project management tools to identify dependencies among several tasks and workstreams before implementation begins is crucial. Not only does this improve project flow, but it also identifies pain points early. If a lack of model capability is consistently the limiting factor, perhaps the company should strongly consider a conversion to a modeling platform poised for LDTI compliance.

To successfully identify and address dependencies, communication is key. Under the vast umbrella of communication, it is vital that management place emphasis on 1) employee education and 2) cross-workstream collaboration. Employees should take time to thoroughly understand the new guidance. This can be accomplished by watching instructive videos or reading detailed outlines such as PwC's In-Depth Manual[2] or the Accounting Standards Update (ASU)[3] itself. Regardless of the means, a strong base will enhance dependency identification abilities. For instance, an employee would not appreciate the important role historical data plays in the market risk benefit (MRB) calculation if they do not learn that the attributed fee must be calculated from inception. Once educated, team members from different workstreams must frequently collaborate to gain understanding of other team's perspectives and ensure teams prioritize tasks that impact other workstreams. Introducing cross-workstream oversight can help to enforce a culture of workstreams joining forces to reach common goals and to prevent anything from "falling through the gaps."

HIT THE CUTOFF MAN: THE ASSUMPTIONS DATABASE

Crack! The ball pops off the hitter's bat, just clears the leaping shortstop's glove and splits the sprinting left and center fielders as it makes its way to the warning track. As the runner on base rounds third and heads for home, the outfielder does not attempt to throw home, but instead targets the cutoff man. By doing so, the cutoff man can make a more accurate throw home as he is much closer. Although a long throw from outfield to home is exciting, it has a low success rate and high risk of error. With each long base hit, this risk reappears, and the cutoff man runs out to mitigate this risk every time.

FASB's new requirements greatly increase the number of assumptions a company must review, update and maintain. Each assumption introduces additional risk. The ball (assumption) is hit to the wall, and it somehow needs to make its way to home plate (embedded in the final required disclosures). We recommend to hit your cutoff man, the assumptions database. An assumptions database helps to address the new implication of having to maintain many more assumptions at granular levels. A few examples are:

- **Requirement.** Periodically update Financial Accounting Standards (FAS) 60[4] assumptions for reserve unlocking
- **Implication.** Rather than maintaining one set of locked-in assumptions, store multiple sets of assumptions for each unlocking valuation date
- **Requirement.** Update the liability for policyholder benefits (LFPB) discount rate every quarter, but calculate its underlying net premium ratio (NPR) using its original discount rate from inception or transition date
- **Implication.** Store original curve—either a forward curve, spot curve or average rate depending on company's account policy decisions—and curve at each quarter moving forward in order to build required LFPB disclosures and rollforwards
- **Requirement.** Calculate an attributed fee percent (AF%) at inception for each contract with an MRB
- **Implication.** Store this locked-in AF%, which equals the ratio of the fair value of expected MRB benefits to the fair value of expected total fees, at a seriatim level for every single policy containing an MRB

Having one source of assumptions will mitigate risks of errant handoffs from assumptions teams to modeling teams to valuation teams. This can also enhance process efficiency, as maintaining all assumptions in a consistent structure period over period allows companies to automate the process of updating the database and feeding the models.

These assumptions will also be applied at different levels. An LFPB may be calculated at a cohort level that was determined based on issue year, while an MRB will be calculated at the policy level. Thus, assumptions must not only be maintained at a granular level but also map to higher level groupings such as cohorts.

PLAY BALL!

The end of the game may seem far away as we sit in the dugout during this rain delay, but the final pitch will be here before we know it. In the meantime, if your company gets the signs right and hits a few cutoff men, along with a few other things, it can enjoy some great success:

- Modernizing an end-to-end process, ultimately increasing efficiency and effectiveness
- Complying with FASB LDTI regulations by the deadline while establishing strong controls and management practices for the future
- Learning from project challenges and pitfalls to develop better practices and strategies for future endeavors

If your company can accomplish that, it will have crossed home plate with the winning run. But for now, it's time to come out of the dugout, play ball and start to modernize because it won't be raining for long. ■

Tim Koenig, FSA, MAAA, is a senior associate at PwC based out of Philadelphia. He can be reached at *timothy.koenig@pwc.com*.

ENDNOTES

1 PwC. 2019. FASB Proposes Standard Deferral of Long-Duration Standard. Aug. 21. *https://www.pwc.com/us/en/cfodirect/issues/insurance-contracts/fasb-deferral-long-duration-standard.html*

2 For a video and outline, see PwC. 2018. Detailing the New Accounting for Long-Duration Contracts of Insurers. May 16. ***https://www.pwc.com/us/en/cfodirect/publications/in-depth/long-duration-contracts-asu-2018-12.html***.

3 Financial Accounting Standards Board (FASB). 2018. Financial Services—Insurance (Topic 944): Target Improvements to the Accounting of Long-Duration Contracts. *Accounting Standards Update*, no 2018-12. *https://www.fasb.org/jsp/FASB/Document_C/DocumentPage?cid=1176171066930&acceptedDisclaimer=true*.

4 Financial Accounting Standards Board (FASB). 1982. Statement of Financial Accounting Standards No. 60: Accounting and Reporting by Insurance Enterprises (FAS 60). *https://www.fasb.org/jsp/FASB/Document_C/DocumentPage?cid=1218220127121&acceptedDisclaimer=true*

Insights Into Life Principle-Based Reserves Implementation and Modeling Practices

By Kevin Carr II, Simon Gervais, Haley Jeorgesen and Chris Whitney

Mandatory implementation of life principle-based reserves (PBR) is just around the corner and there is no shortage of work to do, as most products have yet to be moved to PBR.

Oliver Wyman recently completed its 2019 PBR survey, with more than 40 participants covering 85 percent of the individual life market, including 23 of the top 25 life writers and five reinsurers. This article will expand on the key survey findings shown in Figure 1, elaborating on implementation trends, analysis to date, model robustness and common simplifications.

Figure 1
Survey Key Findings

Analysis to Date

- PBR implementations are heavily back-loaded, with 75% of participants' products moving to PBR in Q3 2019 and later.
- PBR implementations were light in 2018 as compared to 2017, perhaps indicative of the effort required to support products transitioned in 2017.

Actuarial Models

- Only 25% of participants have integrated asset-liability models for PBR, driving widespread use of simplifications related to asset modeling.
- Liability modeling capabilities are more advanced than assets; the most common simplification being the exclusion of riders.

PBR IMPLEMENTATIONS ARE HEAVILY BACK-LOADED

Figure 2 shows actual PBR implementations through 2018 and planned implementations through the remainder of the optional implementation period.

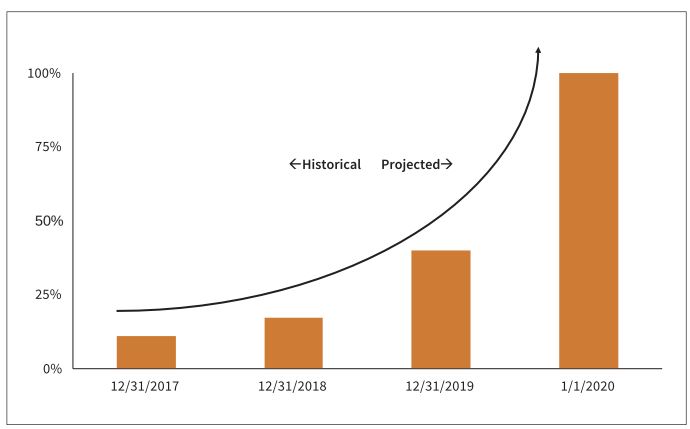
Aside from an influx of products moved to PBR in 2017, few products have been moved to PBR during the optional three-year phase-in period. Planned implementations for 2019 will primarily occur in the fourth quarter, followed by an influx of the remaining products at the start of 2020.

We believe the back-loading of PBR implementation is driven by the following:

- Competitive pressures and prevalence of reserve financing solutions for term and, to a lesser extent, universal life with secondary guarantees (ULSG), for which reserve reductions decrease tax leverage
- Resource constraints and the level of effort required to move products to PBR, including additional reporting and disclosure requirements
- Evolving PBR requirements, which have material impacts on profitability

Keeping implementation timelines on track will be crucial in the final stretch of the optional phase-in period, which is becoming increasingly difficult as preparations for other regulatory changes (e.g., Financial Accounting Standards Board targeted

Figure 2
Percentage of Products on PBR



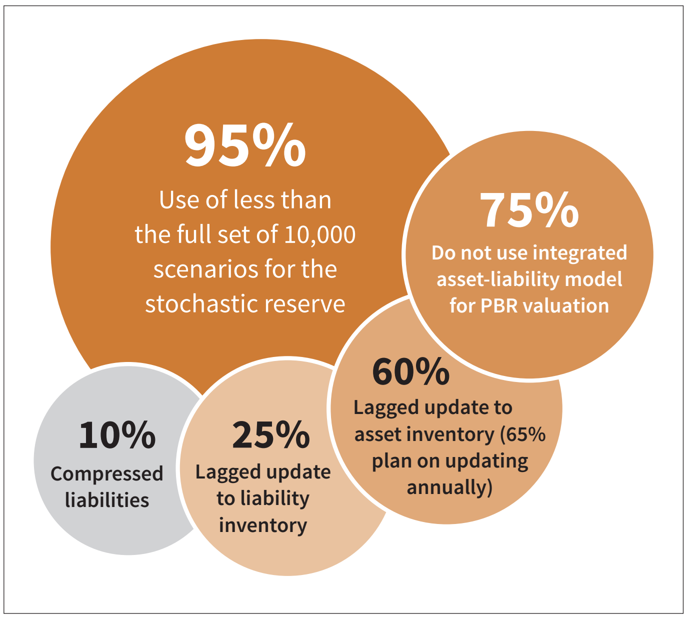
Note: The percentages were calculated as (number of products on PBR)/(total products expected 1/1/2020).

improvements for long-duration contracts, variable annuity reform, International Financial Reporting Standards updates) are underway.

PBR MODELING SIMPLIFICATIONS REMAIN WIDESPREAD

In light of the considerable pressure from accounting and regulatory changes on modeling teams and heavy backloading of PBR implementations, PBR modeling simplifications remain widespread, especially with regard to assets. See Figure 3 for additional details on common simplifications.

Figure 3
PBR Model Robustness and Simplifications



Model simplifications related to assets and scenarios are most prevalent, as this is the area where PBR required most new functionality.

The most common model simplification for liabilities was the exclusion of riders from modeled reserves,[1] as shown in Table 1.

Table 1
Treatment of Riders in Modeled Reserves

Rider	Exclude
Waiver of premium	80%
Other riders and supplementary benefits	59%
Acceleration of benefit (non-zero-cost)	37%
Long-term care	34%
Acceleration of benefit (zero-cost)	32%

THE FINISH LINE?

Mandatory PBR implementation is upon us, and many products remain to be moved to PBR by Jan. 1, 2020. As stated, we believe the back-loading is largely conscious, but that many implementations are effectively behind, requiring additional focus and resources to reach the finish line.

The extent of model simplifications indicates that many carriers are taking a "smart compliance" approach where they try to leverage existing infrastructure to meet the PBR implementation deadlines, in effect deferring necessary model and process improvements until after the mandatory implementation date.

As the finish line approaches, it is important for companies to skillfully manage the regulatory and accounting changes in order to be prepared and accurate on "day 1," while also establishing a modeling and reporting foundation that is sustainable. ■

Kevin Carr II, FSA, is a senior consultant at Oliver Wyman, located in Hartford, Connecticut. He can be reached at *Kevin.Carr@OliverWyman.com.*

Simon Gervais, ASA, is a consultant at Oliver Wyman, located in Hartford, Connecticut. He can be reached at *Simon.Gervais@OliverWyman.com.*

Haley Jeorgesen is a consultant at Oliver Wyman, located in Atlanta. She can be reached at *Haley.Jeorgesen@OliverWyman.com.*

Chris Whitney, FSA, MAAA, is a principal at Oliver Wyman, located in Hartford, Connecticut. He can be reached at *Christopher.Whitney@OliverWyman.com.*

ENDNOTE

1 Deterministic reserve (DR) and stochastic reserve (SR).

Confessions of an Efficiency Junkie

By Jeff Samu

Throughout my career, I've attended many all-employee meetings. And in almost all of those meetings, I heard about how important efficiency is:

- "We must get more efficient!"
- "Our goal is continuous improvement!"
- "Process improvement is a major key to our future success!"

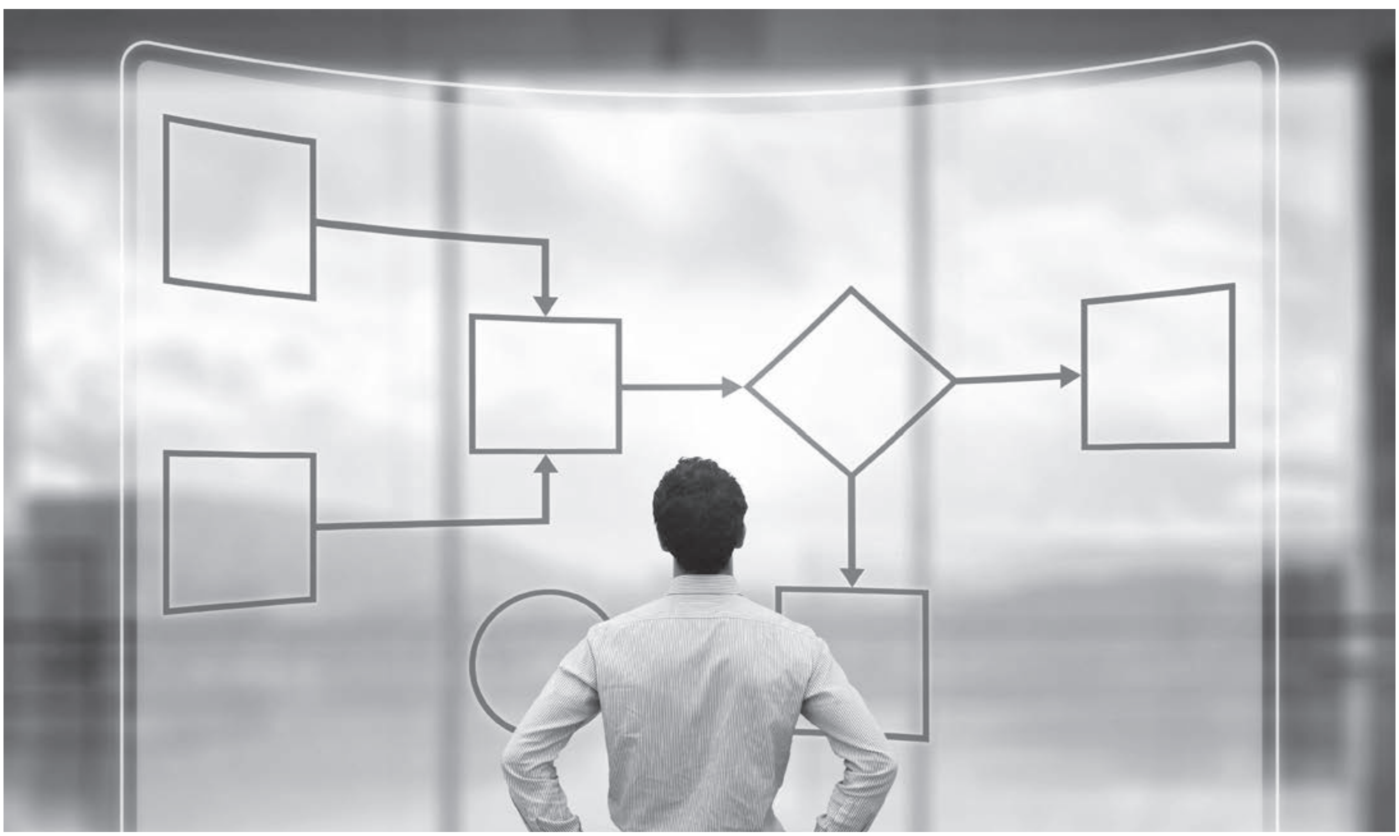
The orders were always clear: In order to succeed, everyone would need to become more efficient. Armed with this directive, I would set out to see where I could improve efficiency. In most cases, I would look at a specific process that I was responsible for and think about where the bottlenecks were. I made a habit of rebuilding my processes, introducing macros to eliminate manual steps and make them run faster. I developed strong coding skills to generalize the macros and allow them to be applicable for broader uses. I started playing around with Office libraries and application programming interfaces (APIs) to enable my programs to talk to others without my having to pass the information back and forth. I expanded my reach beyond my own processes and developed tools that were helping others to automate their work. Every day, I was making the company more and more efficient.

Or so I thought.

WHAT IS EFFICIENCY?

I had never really questioned what it meant for something to be more efficient. In my mind, an efficient process was one that ran fast when I clicked the button. As long as I was improving runtime, I was making things more efficient. I could take a process that would take a week of manual data preparation and condense it down to 15 minutes of machine time. I could click a button, go to lunch and return to find my custom-built digital servant ready with my results.

Then, one day, I ran into an issue. One of my beautiful time-saving processes had crashed. At first, I was confused. What



could have possibly happened? It had worked well the previous time I had run it. I combed through the code and found nothing that would raise an alarm. I checked the inputs and those looked good as well. I couldn't understand what went wrong.

Eventually, I discovered the source of the issue. One of our Excel source files had a field that had contained projection dates. In previous quarters, these were entered as text (e.g., 01/01/2019). In the current quarter, the field looked identical, but Excel was actually storing the date as a number (43466). I changed the code for the process so that it interpreted the date correctly, and the process ran without issue.

Except, the next quarter, the format changed back. Since I had updated the code the previous quarter, the process was crashing again. I updated my code again to include a check to see how the date was being presented, and to process it correctly regardless of the presentation.

Meanwhile, I had another problem. I had submitted my results from a separate process to management and was receiving a lot of questions about the numbers. Again, I was confused. My efficient, automated process shouldn't have made mistakes. I reviewed the inputs and noticed a row had been inserted in one of the source files. My process was structured to take a number from a certain cell of the workbook, but that number no longer meant what it had in the past, and the number I was now pulling was different by a factor of 10. I coded some additional logic to allow for more flexibility in the inputs—to look for data labels rather than blindly pulling a value from a particular cell in the workbook.

Then the phone rang. One of my business partners in a less-technical area had a question about a process I had helped him with. It was working fine, but he wanted to make an enhancement and didn't know how to go about doing it. He could follow the code to some extent but wasn't sure where to begin to make the changes he wanted. I realized that each time he wanted an enhancement, he would come to me and I would need to make every update because my ultra-efficient code, which ran as fast as lightning, was hard for anyone else to understand and modify.

These experiences showed me that my initial view of process efficiency had been rather rudimentary. I had a localized view of efficiency: I wanted my components of a process to run fast and structured them accordingly. I realized I had made a number of faulty assumptions in my process design. I had assumed that:

1. My inputs and outputs would be static.
2. Automation was always the best approach.

> At its core, efficiency is about trying to get the greatest value from limited resources.

3. The controls in place were the only ones needed.
4. Any solution I built would always be the best one.
5. Others had the same technical skills I did, and they would be able to maintain the processes.
6. A single program used for different areas and needs would be more efficient than individual programs.
7. Any incremental gains to any process were worth pursuing.

Ultimately, I had assumed a definition of efficiency that I was starting to question.

So then, what *is* efficiency? When management says they want something to be more efficient, what are they really asking for? At its core, efficiency is about trying to get the greatest value from limited resources. It's about building solutions that can answer specific business questions so actions can be taken, without requiring an onerous amount of work. Fundamentally, it's about trying to achieve the best balance of quality, resources and time.

QUALITY

The first element of efficiency, and the most important one, is quality. If a process doesn't help me answer the fundamental question that I'm asking, it cannot be efficient. If my valuation software can't calculate a reserve properly, or if my projection model can't account for a benefit that 90 percent of the in-force policies have, or if I'm developing a hot new product feature I can't price, I need to search for (or build) a solution that can.

That is not to say the quality needs to be perfect. Valuing an exotic product feature that is no longer available and which only 10 clients have may not be necessary. There are many good reasons why model compression or simplifications may be used, and as long as these don't materially affect the results, solutions that use them can still be considered high quality and perhaps even more efficient than ones which do attempt to model everything.

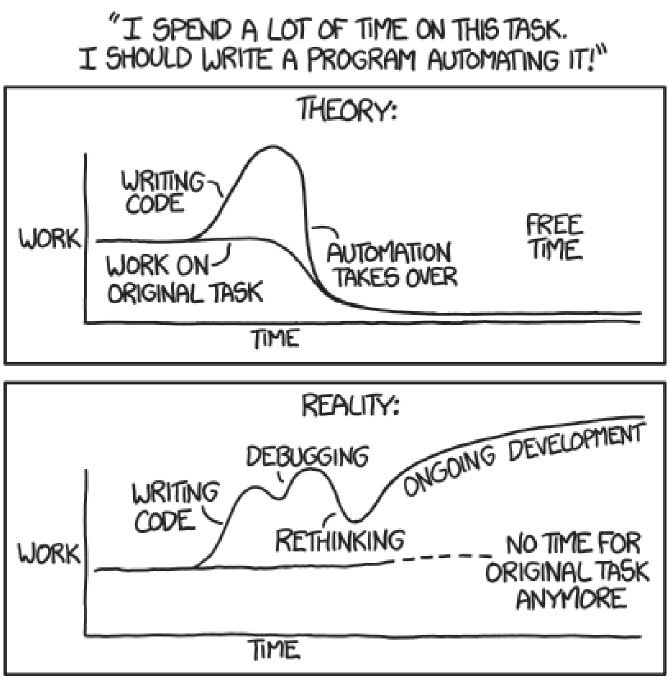
Controls are an integral part of assuring quality. It is critical that effective controls be in place to ensure the results are reliable. These can include validation of the input data, logs of the input files, logs of errors and reports with key values throughout the process. Without these controls, the end result can be suspect, which can lower the ultimate efficiency, as more time needs to be spent reviewing the results.

Quality should be considered on an end-to-end basis, from the initial inputs to the final reports. If my modeling software has some limitations, but I have a post-valuation process (PVP) that can account for these limitations, then the quality of the process as a whole can still be good. A caveat to this is that handoffs between different components in a process can introduce additional risk. In this case, data may be interpreted differently between the components and issues may be more difficult to trace, which can render the overall process less efficient unless appropriate controls are in place.

RESOURCES

When most managers think of resources, they usually think of the number of people involved, and for good reason. Personnel costs are generally the greatest component of a department's budget, and it is generally much harder and more expensive to add or shift people than it is to add technology resources.

When considering efficiency, though, it's important to consider not just how many people are involved, but also what their skills are and what value they are bringing to the process. If a process requires a very specialized skill to build, it will be more difficult to find someone to build it, more expensive to implement it and more challenging to maintain it. There may be only a few hundred actuaries who are intimately familiar with a given modeling platform, but there are thousands of actuaries who can understand Excel, and hundreds of thousands of programmers who are fluent in C++, Java, SQL or VB.net.



Source: Munroe, Randall. Automation. Jan. 20, 2014. *https://xkcd.com/1319/*.

This variation in skills becomes more critical when the users and developers are different. If users can't understand the process, they won't be able to maintain it, and the developers' specialized knowledge will be needed for updates. Over time, this can result in the developers solely supporting existing applications and not being able to build new solutions, constricting their ability to address bigger issues and reducing their overall value.

Resource availability and opportunity cost are often overlooked. When one of my customers asked for a particular fix to a process I was supporting, I remarked it would take an hour to fix, but it would take me two months to find that hour. Every hour I spent on that process was an hour I couldn't spend on something else, and since I was the key technical resource for a number of critical processes, my availability had to be prioritized. As a result, while each individual process may have been quite efficient on its own, the overall efficiency of the portfolio suffered.

TIME

When making a process more efficient, saving time is often the goal. When I would redesign a process, my goal was often to reduce runtime by a factor of 10 or more. If it took a day to do before, I wanted to get it down to less than an hour.

But runtime should not be the only time considered. The measurement of time should be all-inclusive, incorporating runtime of a calculation engine with time spent in design, development, debugging, testing, enhancements, review and reporting. Suppose I have a monthly process, and I reduce the runtime by an hour each month, but it took 20 hours to code the new process, five hours to test and debug it, and an extra 10 minutes to review the inputs and results each time to make sure the process didn't miss anything. Investing that time may not give the desired payoff and may even reduce the overall efficiency.

Furthermore, the amount of time isn't as important as the value of that time. The value of time is not the same for each person and can change over time. The value of time for an entry-level actuary is quite different from that of a chief actuary. A

Source: Munroe, Randall. Data Pipeline. Oct. 3, 2018. *https://xkcd.com/2054/*.

valuation actuary would likely spend 10 hours in the middle of May to develop something that saves one hour in January. Opportunity cost needs to be considered in these valuations as well, as the same time cannot be spent on multiple projects, so the relative value of the projects will influence the value of time spent on them.

CONCLUSION

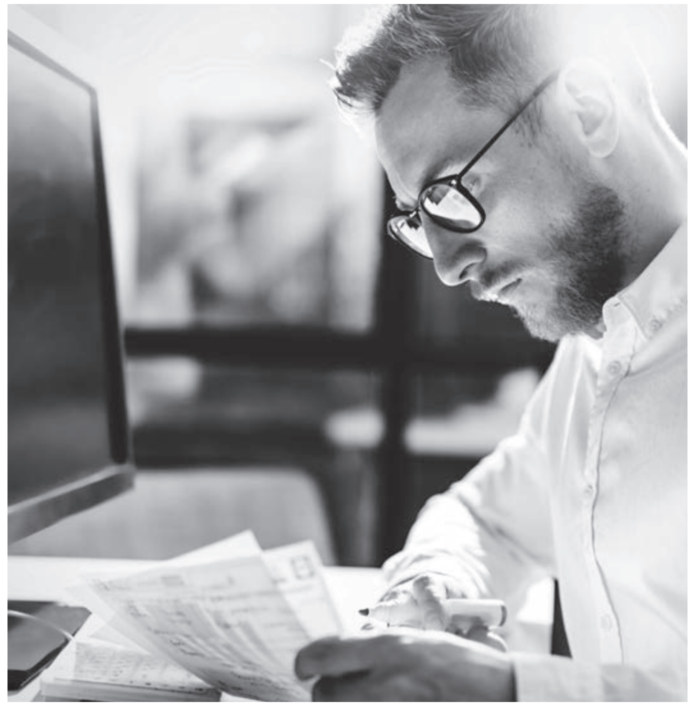
When I had considered efficiency in the past, I always thought of it as an exercise in improving runtime through automation. Today, I realize it's not nearly that simple. Efficiency is about balancing quality, resources and time in a way that makes sense for that particular application. There is no universal approach to efficiency that works for all situations. But by considering the requirements of a process and the conditions of the resources, you can make intelligent steps toward making your process more efficient. ■

Editor's note: For a deep dive into the process assumptions, see the expanded version of this article on The Modeling Platform *digital edition at* https://sections.soa.org/view/society-of-actuaries/the-modeling-platform.

Jeff Samu, FSA, MAAA, is a vice president and actuary in the Business Architect group at Prudential Financial. He can be reached at *jeffsamu@hotmail.com.*

Writing Effective Model Documentation

By Dodzi Attimu, Ryan Bijur, Jeffrey Jardin, Michael Minnes and Veltcho Natchev



The main objective of technical documentation is to minimize key person dependency for the model users/owners and to allow any reasonably competent modeler (including a new one) to understand its methodology. The guiding principle of effective model documentation is to provide a description of the model's methodology and functionality and their implementation in support of the model's objectives and requirements.

Comprehensive documentation should contain sections on model inputs, calculations, outputs, limitations, associated business processes, governance practices, application and platform specifics. Documentation challenges arise with the operational components of a model: input structure, throughput and output structure.[1] Consequently, this article will focus on the documentation of the components:

- **Inputs.** Data, assumptions and parameters
- **Throughputs.** Model theory and calculations
- **Outputs.** Interim calculated variables and final model results

As much as possible, documentation of methodology should be platform, application and code neutral in the sense that formulas should be generic rather than reflect the programming language. This way, if there is a model conversion to an entirely different system, the documentation will not have to change (given that the methodology is the same/has not changed).

While focusing on content, it's important to remember that model documentation should be easy to maintain and update. For example, a common approach is to use lists and tables as centralized information storage units, potentially placed in an appendix, rather than creating lengthy descriptions in the body of a document. The documentation should reflect the model as it is, rather than how it should be, eliminating judgment by the documenter with the presumption that the model is working correctly.

Our approach to documentation is to use a single universal template stored in a central repository. The benefits of this include comprehensive coverage of model elements, documentation style consistency, information security and ease of use.

INPUTS

All models use inputs such as data (in-force extracts, scenarios, etc.), assumptions and parameters. It is important for the model documentation to have a detailed description of the input facility. A description of data derivation external to the model is not needed; however, there should be links to data sources.

A recommended approach would contain these steps:

- Specify the input structure and data elements (including assumptions) and sources, how they are entered or uploaded/downloaded into the system (model), for example, manually, via automated processes or combination of both

- Describe how inputs are changed and how the changes are tested

- Provide a detailed description of the input facility for the model, such as groups, tables, variables; these should include naming conventions

Inputs should be tied to existing model artifacts, such as assumption memos. This enables the user to understand how the assumptions are coded/used in the model. If the model uses an open system (homegrown or third-party provided open-source code software), variables that read the input data should be listed.

For variables calculated using the inputs and then used for further processing within the same model, such as dynamic lapses or experience mortality, it would be appropriate to include them in the Input and Calculations sections. References to the calculations should be included where appropriate.

Input structures that support sensitivity/stress-testing should also be documented.

THROUGHPUTS (CALCULATIONS)

The calculations within a model reflect its theoretical underpinnings, and good documentation will not only present the formulas used in these calculations but also provide insight into the background of modeling simplifications and methods used. However, there is a trade-off between verbosity and accuracy: The more detailed the documentation, the higher the likelihood of unfaithfully representing its functionality.[2]

To effectively describe the model calculations, it is useful to:

- Describe how the model obtains its output. For instance, does the model aggregate cells from a seriatim projection? Does the model aggregate individual model cells? Does the model project a number of stochastic scenarios and determine the mean of a certain value?

- Outline what the model is projecting or representing along key dimensions (e.g., starting time, horizons, scenarios, elements that are being projected) in support of each of the model's uses (objectives, purposes and business processes). It is important to also clarify the order of operations, such as assessment of charges and crediting of interest.

- Decide whether to use a top-down or bottom-up approach to document the calculations, using the expertise, experience and discretionary knowledge of the documenter.

Example: Cash Flow Testing

To clarify the calculations, any aggregation used to produce the final output should be described (with formulas, if appropriate) and the core calculations should be documented for a representative "cell(s)" used in the aggregation variations in modeling for different cells should also be shown.

- It may be necessary to set the notation for the consecutive time variables $\cdots, t_{k-1}, t_k, t_{k+1}, \cdots$, for example, as follows:

  - In many instances, the beginning of period t_k is conceptually synonymous with the end of period t_{k-1}.

  - Another related consideration is the order of operations in a given time (period).

- Statutory surplus is calculated at each time t_k, for each Scenario ξ, for a policy as:[3]

  - Surplus(t_{k+1};ξ) = ProductRevenue(t_k;ξ) + NII(t_{k+1};ξ) – ΔReserves(t_k,;ξ) – Benefits(t_{k+1};ξ) – Expenses(t_k;ξ).

- In some instances, it may help to specify the order of operations utilizing, for example, the following:

  - $t_k \equiv$ beginning of period t_k
  - $t_k^- \equiv$ end of period $t_{(k-1)}$
  - $t_k^+ \equiv$ an instant after beginning of period t_k post beginning of period deductions

  Or generically in words, for example: Fees are always deducted prior to calculation of investment income …

The next level of detail would involve documenting the individual components of the previous equation, including ProductRevenue($t_k$$\xi$), NII($t_{k+1}$;$\xi$) and ΔReserves($t_k$,;$\xi$). The components may need further documentation of their subcomponents until one gets to an appropriate level of detail, the lowest level being the input structure/variables. It is important to document the relationship between the input data structure and the variables that show up in the formulas as needed to clarify such relationships.

- Note the use of variables rather than concrete values—in the example, the time steps were denoted $\cdots, t_{k-1}, t_k, t_{k+1}, \cdots$. Of practical importance are the intervals between time steps, $\delta t_k := t_k - t_{k-1}$, as well as the last time step, t_N.

- Using the variables this way makes the document easier to maintain because, even if the current implementation consists of monthly projection time-steps for 40 years, one section of the document could mention that the current implementation has δt_k to be monthly with $N = 12 \times 40$. Other potential nuances like calender month or policy month consideration could be handled similarly.

This example utilizes a top-down approach, where we start from the output variable(s) being documented and work progressively down to the underlying components. Alternatively, one could utilize a bottom-up approach starting from the lowest level of detail and working up to the ultimate output variable of interest. In that case, one would start from the mortality assumption input through persistency assumptions to the components of the core variable. In addition, this approach is particularly useful when documenting multiple outputs resulting from a common set of core calculations. Ultimately, the documentation could utilize a hybrid approach consisting of both the bottom-up and top-down methods.

Further Considerations

A key methodology documentation challenge arising with vendor systems is the potential need to rely significantly on vendor documentation. Therefore, especially for closed or semi-closed systems, the model documentation should cover the specific/custom configuration/implementation by the user. In addition, for these closed systems it is important to:

- Create appropriately detailed description of any calculations performed on the inputs and assumption data used to populate the model.
- Explicitly describe how the assumptions are intended to be populated. For example, a pricing model may contain distinct mortality/morbidity assumptions for projecting benefits and reserving. The documentation should make it clear how each of these assumptions flow into the ultimate calculations to help the user understand how the product is modeled. This can be done by describing the calculations, when known, or by providing links to the vendor's online help.
    - Provide a description of calculations with references to the vendor-supplied documentation; if the latter is not complete, provide formulas compliant with corresponding regulations, Actuarial Standards of Practice and so on.
    - Indicate and describe, without necessarily attempting to reverse-engineer the calculations, which methods are being used (i.e., what is the model's supporting theory) and be very explicit around **core** calculations. Often the business unit that uses a closed system will have spreadsheets or small programs used to test some key model calculations. These can often be referenced to illustrate a calculation and should be included in the documentation appendix.

OUTPUTS

Model output, as defined earlier, is generally a collection of tables populated by the calculation engine. For example, model output can consist of files, external reports or reporting facilities within the software. Usually third-party vendor-supplied systems have more developed reporting facilities, pre-packaged reports and, typically, customized reports, as well. Non-vendor provided (homegrown) applications are usually custom-built for the business. The model's output should be documented, but external reporting facilities that perform further manipulations on the output can be documented elsewhere.

Any output structure generated by the model should be documented:

- Similar to the input, the documenter can use a data dictionary to describe the output structure.
- At a minimum, the data dictionary should contain the following two components:
    - The description of line items and
    - The formulaic expressions that relate them to other variables, if applicable, provided they are not documented in the calculation sections.

Table 1
Example of Output Structure Documentation

Field Name	Field Description	Notes
t	Time step	This is end of month or end of year depending on $\delta t_k := t_k - t_{k-1}$, setting of the projection time steps
. . .	. . .	. . .
InvIncome(t)	Income from investments: coupons, principal etc.	Documented in the calculation section
RCGL(t)	Realized capital gains and loss	Documented in the calculation section
InvExp(t)	Investment expenses	Documented in the calculation section
NetInvIncome(t)	Net investment income	= InvIncome(t) + RCGL(t) InvExp(t)

- Each output report may correspond to an (output) structure, as shown in Table 1.

In some cases, there may be multiple outputs, in one or multiple tables, corresponding to different scenarios/sensitivities. It is helpful to document these as well. In either case, it is necessary to provide a comprehensive description of outputs, including key variables, calculations generating the outputs and reserving. Other purposes of the output such as accounting basis and financial statements, should be specified. The descriptions provide context and make the documentation understandable to someone unfamiliar with the model. These details enhance the modeler's ability to maintain the model.

CONCLUSION

In this article we focused on three items related to technical model documentation. Proper documentation reduces key person risk, decreases a new modeler's learning curve, provides a consistent standardized companywide process, and helps perform corporate audits, deep dive validations and model conversions.

Comprehensive documentation should also include other items, such as model objectives and limitations, control structure, testing, change management and other relevant information. Many of these are not just informative for the modelers but may constitute a requirement from various governing bodies, such as enterprise risk management, internal and external audit and rating agencies. As a result, the final documentation product becomes a powerful and valuable store of institutional model knowledge base. ■

Dodzi Attimu, FSA, CERA, CFA, MAAA, Ph.D., heads the model validation program at MassMutual. He can be reached at *dattimu06@massmutual.com*.

Ryan Bijur, FSA, MAAA, is an actuary at MassMutual. He can be reached at *rbijur28@massmutual.com.*

Jeffrey Jardin, FSA, is a model risk management actuary at MassMutual. He can be reached at *jjardin61@massmutual.com.*

Michael Minnes leads model governance within enterprise risk management at MassMutual. He can be reached at *mminnes@massmutual.com.*

Veltcho Natchev, FSA, MAAA, is a model validation actuary at MassMutual. He can be reached at *vnatchev@massmutual.com.*

ENDNOTES

1 Attimu, Dodzi. 2018. What's a Model? A Framework for Describing and Managing Models. *The Modeling Platform* November. *https://www.soa.org/globalassets/assets/library/newsletters/the-modeling-platform/2018/november/mp-2018-iss8.pdf*.

2 For example, the statement "This cash flow test model projects all relevant cash flows for the book of business" is technically accurate, but does not provide useful information. The model documentation should provide sufficient detail into its calculations as it is important to accurately capture the model calculations without being too high level. However, once we start adding details of the calculation risks of misrepresentation may arise. Sometimes it means adding a caveat to certain aspects to point out the existence of further lower-level complexity.

3 The model produces aggregate results over the policy calculations that follow.

Key Principles of Actuarial Model Governance

By Mitchell Stephenson

In the decade following the financial crisis, actuarial model governance programs and guidance have become increasingly prevalent. According to a 2012 report on actuarial modeling controls by the Society of Actuaries (SOA), approximately one-half of the survey respondents did not have a formal, written policy that governed the modeling process.[1]

Just five years later, that number had increased significantly to 75 percent.[2]

There are several reasons for the recent increase in actuarial model governance programs and guidance:

- Models were perceived to be ineffective in producing sufficiently severe outcomes during and after the financial crisis.[3]
- Increased computer power and data, sophisticated modeling techniques and complex modeling software has increased model risk.[4]

The increase in model governance programs points to an industry awareness of the impact these programs have in mitigating model risk. In the future, as artificial intelligence and big data analytics become more prevalent and calculations more complex, it will be critical that individuals using these tools have confidence the results are correct. A strong actuarial model governance program gives model owners and users confidence that their models work correctly. As a result, they can spend less time troubleshooting potential model errors, and more time on insightful analytics and communicating results.

A sound actuarial model governance program that abides by these key principles will help an organization to maintain, and ensure, this necessary level of confidence in its modeled results.

There is ample guidance available to actuaries on the topic of model governance.This includes practice notes and checklists by the International Actuarial Association (IAA) and American Academy of Actuaries (AAA), references in Actuarial Standards of Practice (ASOPs), and guidance by actuarial organizations such as the SOA, Casualty Actuarial Society (CAS) and the North American Chief Risk Officer (CRO) Council. This guidance can help organizations ensure they are following best practices to mitigate model risk.

Much of the guidance, however, only covers a specific aspect of model governance, like the North American CRO Council paper on model validation, or offers detailed guidance of considerations, like the AAA checklist on model governance.[5] It is difficult to derive the key principles from the large amount of available guidance, which varies in depth and content. To help understand the guidance, here are the key principles of model governance that practicing actuaries should be aware of:

- **Sustainability.** A formal structure and written policy that includes defining what a model is, a process for keeping a model inventory and a clear segregation of duties—including independence in model validation—will create a sustainable governance program. This written policy will encourage consistency, reduce ambiguity and ensure the governance program will "live on" beyond the current team working with it.

- **Third-party principle.** The third-party principle ensures that someone reviewing model documentation, like a future model developer, user, validator or auditor, can generally understand what the model is doing and why, especially when it comes to methodology decisions or simplifications. Oftentimes, documentation is too long and complex for others to find relevant information, or so short that others cannot glean an understanding about the model. A model's documentation should clearly cover its intended use, inputs, calculations, outputs, use limitations, and the results of testing and validation.

- **Integrate governance into the process.** Integrating model testing and documentation best practices into the model development life cycle will encourage efficiency by reducing the time needed to separately perform these activities after the model is fully developed. Integrating testing and documentation also encourages consistency by embedding these best practices into a repeatable process.

- **Accountability.** Having an accountable party to ensure there is a formal model governance structure is the best way to ensure the right activities take place. Recent ASOPs point to the actuary as being accountable for ensuring reasonable governance and controls around model development and use. This is true for pricing models as addressed in ASOP 54, valuation models as addressed in the Valuation Manual sections VM-G and VM-30 and ASOP 52, and modeling actuaries as addressed in the Modeling ASOP Exposure Drafts.

- **Risk-smart.** To ensure the right amount of effort is spent on the right activities, materiality and risk should be applied to the testing and validation of model inputs, calculations and outputs, as well as to sensitivity testing and model documentation activities. Actuaries need to recognize insignificant or immaterial parts of this process and spend more time on impactful risks.

Actuarial work, like many other professions, will continue to be impacted by new advances in technology. The "actuary of the future" will likely spend less time on data-driven exercises and more time on explaining and communicating results through insightful analytics. To be prepared for this future, actuaries must have confidence that the models work as intended and are appropriate for their intended use. A sound actuarial model governance program that abides by these key principles will help an organization to maintain, and ensure, this necessary level of confidence in its modeled results. ■

Mitchell Stephenson, FSA, MAAA, is the head of Actuarial Model Management and Controls at Prudential Financial. He can be reached at *mitchell.stephenson@prudential.com.*

ENDNOTES

1 Kaufman, Sara V., Jeffrey R. Lortie and Jason A. Morton. 2012. Actuarial Modeling Controls: A Survey of Actuarial Modeling Controls in the Context of a Model-Based Valuation Framework. Society of Actuaries report. *https://www.soa.org/globalassets/assets/Files/Research/Projects/research-2012-act-mod-contr.pdf.*

2 Chamberlain, Thomas, Charles Deak, Katie Egan, Sara Viet Kaufman and Jason Morton. 2017. Actuarial Model Governance: A Survey of Actuarial Model Governance and the Industry Evolution. Society of Actuaries report. *https://www.soa.org/Files/Research/Projects/2017-01-actuarial-modeling-controls-research-report.pdf.*

3 North American Chief Risk Officer (CRO) Council. 2012. Model Validation Principles Applied to Risk and Capital Models in the Insurance Industry. *http://www.crocouncil.org/images/CRO_Council_-_Model_Validation_Principles.pdf.*

4 International Association of Actuaries. 2010. Note on the Use of Internal Models for Risk and Capital Management Purposes by Insurers. *http://www.actuaries.org/CTTEES_SOLV/Documents/Internal_Models_EN.pdf.*

5 American Academy of Actuaries. 2016. Model Governance Checklist: Some Considerations for Life Practicing Actuaries. *http://dev.actuary.org/files/publications/PBRChecklist_Final.pdf*

Economic Scenario Generators, Part I: Motivation for Stochastic Modeling

By Rahat Jain, Dean Kerr and Matthew Zhang

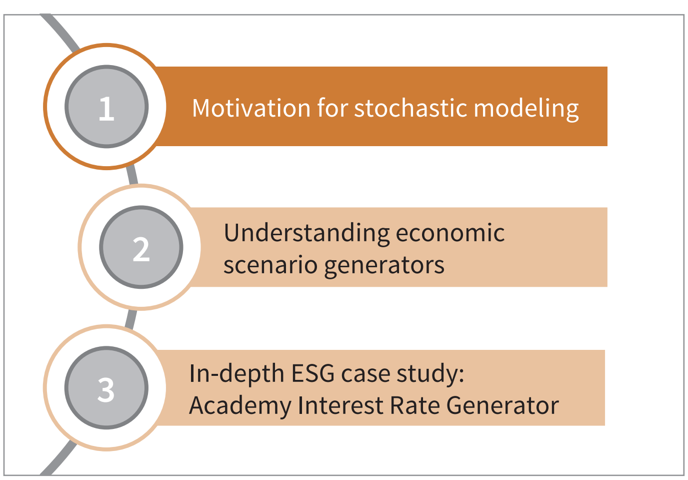
Stochastic modeling has gained increasing relevance in life insurance in recent years, driven by regulatory changes and other factors. Consequently, the use of economic scenario generators (ESGs) by actuaries is becoming ever more common.

This article is the first installment of a three-part series that aims to provide an overview of ESGs and will focus on the general motivation for stochastic modeling, its advantages and its limitations. Future articles will further break down the key factors underpinning ESGs and relate these to the American Academy of Actuaries Interest Rate Generator (AIRG), the most commonly used ESG by U.S.-based actuaries. See Figure 1 for an overview of upcoming articles in this series.

WHAT IS STOCHASTIC MODELING?

Stochastic modeling simply refers to modeling with a random variable rather than purely pre-defined assumptions; it is a powerful tool used in many fields from biology to cryptography. Applications in finance and actuarial science focus on representing seemingly random behavior for factors such as asset returns, exchange rate movements or interest rates.

Figure 1
ESG Three-Part Series Structure



Actuarial models consume information about the past, which is known, and assumptions about the future, which is estimated, in order to project potential outcomes. Predicting the future can be achieved through a single deterministic track or by iterating through multiple potential outcomes. ESGs play a key role in stochastic modeling by simulating future paths of economic and financial outcomes.

ADVANTAGES OF STOCHASTIC MODELING

Stochastic modeling of certain key assumptions can have significant benefits over deterministic methods, as illustrated in Figure 2.

Figure 2
Stochastic Modeling Advantages

1. Measuring "Tail Risks"

Stochastic modeling is a valuable tool for quantifying the extreme events that may arise from market and economic volatility. Unlike traditional actuarial risks, these exposures are generally not diversifiable. Stochastic methods allow for the likelihood of outcomes to be measured and also provide valuable information about the most impactful risk drivers.

Consider a risk management context where stochastic analysis indicates a company's portfolio cannot survive an equity market shock similar to the 2008 financial crisis. The need to react to and mitigate this risk varies greatly depending on the likelihood of that event occurring. A near-certainty versus a one-in-a-million anomaly will drive very different reactions.

2. Enabling Market-Consistent Valuation

Certain financial reporting frameworks require assets and liabilities to be held at fair value. Given the rise of complex financial derivatives and insurance liability offerings that do not have market-observable prices or closed-form solutions, stochastic modeling techniques facilitate valuation of these complex assets and liabilities in a market-consistent fashion.

Consider the exercise of assigning a market-consistent value to a variable annuity (VA) product with a guaranteed minimum accumulation benefit (GMAB). The future outcomes associated with that offering are tied to a wide range of assumptions (interest rate, general and separate account performance, mortality, lapse, withdrawal, annuitization, etc.) which have complex interactions (consider dynamic lapse assumptions driven by market performance). An actuarial model projecting this product across a large number of risk-neutral scenarios is often the only practical method for market-consistent valuation.

3. Satisfying Internal and External Stakeholder Requirements

Stochastic modeling may not be a preference but rather a requirement. Whether driven by the needs of external or internal stakeholders, the application of stochastic modeling techniques is ever-growing.

The recent push toward principles-based reserving methods for U.S. life insurers has supplanted decades of traditional formulaic methods and introduced additional stochastic requirements.

4. Quantifying Asymmetric Responses and Path Dependency

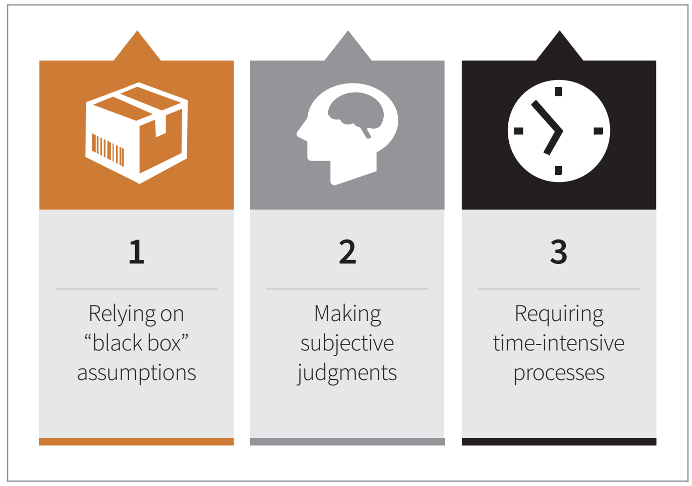
Certain risks cannot be captured adequately by projecting a single outcome. Many product features behave asymmetrically against risk factors or exhibit strong path dependency. This may be exacerbated by dynamic modeling techniques that tie decisions to market outcomes. In these cases, stochastic modeling of many randomly generated paths offers a natural solution to assessing risk.

Consider a put option that has value tied to decreases in a stock price. A traditional deterministic assumption representing "best estimate" does not offer much insight into the risk of issuing such an option—that is, one may erroneously conclude that because the "expected path" of a stock is to increase, selling this option is risk free. To accurately assess the value of such an option, one needs to weigh the likelihood of payoffs across a number of paths.

LIMITATIONS OF STOCHASTIC MODELING

It is important to note that stochastic approaches are one method of forecasting future outcomes; they do not innately represent a more accurate view of the future. The following factors, seen in Figure 3, should be considered when evaluating the use of stochastic modeling.

Figure 3
Stochastic Modeling Limitations



1. Relying on "Black Box" Assumptions

It is common to accept a set of scenario data as a handoff—companies often separate the producers and users of economic scenarios, leading to incomplete knowledge transfer. Scenario users may not have the ability to glean the various assumptions, decisions and compromises that are baked into scenarios from observation alone, potentially leading to misapplication.

Risk-neutral scenarios "look and feel" just like real-world scenarios but serve vastly different interpretations. While there are techniques to differentiate these rather distinct views, an uninformed user can readily draw illogical conclusions such as "in the worst 5 percent of cases, the liability is expected to be valued at X" when, in fact, risk-neutral scenarios are not meant to represent a plausible outcome of future real-world outcomes.

2. Making Subjective Judgments

Stochastic modeling is a projection of the future that can potentially bury biases and expectations. Everything from the choice of ESG, selection of process and parameters, calibration methodology and subjective future outlook drives significant differences in the characteristics of produced scenarios. Coupled with the "black box" nature of economic scenarios, these decisions may not align with the intent of the scenario user.

If two actuaries were asked to project interest rates over the next 10 years, they would more than likely arrive at different answers. Even when the objective is clear, such as generating real-world scenarios and aiming for historical fidelity, a simple decision such as how much historical data to leverage will fundamentally change the outcome. In projecting interest rates, an individual using five decades of historical information, including the higher interest rates of the 1970s, 1980s and 1990s, will have a very different outlook than someone only reflecting the lower interest rate environment of the past decade.

3. Requiring Time-Intensive Processes

Stochastic modeling typically requires significant processing time. Increasing the number of scenarios leads to a direct increase in modeling workloads (i.e., two scenarios will double runtime in the absence of distributed processing).

The large amounts of data being stored to create and model with an ESG, along with the downstream model processes that leverage those scenarios, may stress existing workflows and information technology infrastructure. While advances in computing power and distributable processing options have dampened the impact on processing times, these solutions undoubtedly come at an additional cost when compared to deterministic modeling approaches.

Doubling model runtime in the context of model development would be an inferior outcome. However, processing a model through 500, 1,000 or 10,000 stochastic scenarios fundamentally changes the scale of model runs (e.g., a five-minute run, across 1,000 scenarios, takes more than three processing days). Running stochastic-on-stochastic projections further increases runtime.

CONCLUSION

Stochastic modeling and, by extension, the use of ESGs, is being adopted in many aspects of actuarial work including risk management, hedging, pricing and regulatory compliance. It is essential that model users comprehend the strengths and weaknesses of stochastic modeling and proper application of ESGs.

Stay tuned for the next two installments of our three-part series as we look to discuss the key factors underpinning ESGs and have a closer look at the AIRG, the most commonly used real-world ESG for U.S.-based actuaries. ■

The views or opinions expressed in this article are those of the authors and do not necessarily reflect the official policy or position of Oliver Wyman.

Rahat Jain, ASA, CERA, ACIA, is a consultant at the actuarial practice of Oliver Wyman. He can be reached at *Rahat.Jain@OliverWyman.com.*

Dean Kerr, FSA, ACIA, MAAA, is a partner at the actuarial practice of Oliver Wyman. He can be reached at *Dean.Kerr@OliverWyman.com.*

Matthew Zhang, FSA, CERA, MAAA, is a consultant at the actuarial practice of Oliver Wyman. He can be reached at *Matthew.Zhang@OliverWyman.com.*

An R Package for Experience Studies

By Matthew Caseres

Significant effort is required to transform actuarial data from a raw format into a format that can be used for the calculation of decrement rates or average premiums. The expstudies R package was developed to aid in transforming raw data into assumptions.

This package relies on dplyr, a popular R package with highly optimized algorithms for data manipulation. Functions that had no obvious dplyr implementation were written in C++ for high performance. Using this package on a desktop computer, I have been able to handle millions of rows of data without issue.

The package is open source and is available on GitHub and can be installed using Comprehensive R Archive Network (CRAN) with `install.packages('expstudies')`. The package is in version 0.0.5 so there are still improvements to make. The official package documentation is at *https:/actuarialanalyst.github.io/expstudies/.*

HOW TO READ THIS ARTICLE

There is some R code in this article, in grey boxes. If you don't read the grey boxes at all, everything will make sense from just reading the non-code text so don't worry if you don't know R. The table below the code displays the output from running the code. In this case, "records" is a variable in our environment, and we are displaying its value.

```
records
```

Policy Number	Issue Date	Termination Date	Issue Age	Gender
B10251C8	2010-04-10	2019-04-04	35	M
D68554D5	2005-01-01	2019-04-04	30	F

These records are used for demonstration purposes in this article. We assume a data format with a unique policy number, issue date, termination date, issue age and gender.

CREATING EXPOSURES

The raw data has a single row per policy. For calculations, we would like multiple rows per policy where each row represents an interval of time where the policy was in force. The addExposures function does this. By default, exposure rows are created for each policy year.

```
addExposures(records)
```

Policy Number	Policy Date	Start Interval	End Interval
B10251C8	1	2010-04-10	2011-04-09
B10251C8	2	2010-04-10	2011-04-09
B10251C8	*3*	2010-04-10	2011-04-09

Note: Table has been truncated

ADDEXPOSURES() ARGUMENTS

To allow for greater user control, there are arguments that control the creation of the exposure data frame.

Type

We can partition experience by policy month using the type argument with type = "PM".

```
addExposures(records, type = "PM")
```

Policy Number	Policy Year	Policy Month	Start Interval	End Interval
B10251C8	1	1	2010-04-10	2010-05-09
B10251C8	1	2	2010-05-10	2010-06-09
B10251C8	1	3	2010-06-10	2010-07-09

Note: Table has been truncated

Policy years cross multiple calendar years and we might need to do an analysis filtering by both exact calendar year and policy year. This can be accomplished using type = "PYCY".

```
addExposures(records, type = "PYCY")
```

Policy Number	Policy Year	Start Interval	End Interval
B10251C8	1	2010-04-10	2010-12-31
B10251C8	1	2011-01-01	2011-04-09
B10251C8	2	2011-04-10	2011-12-31

Note: Table has been truncated

There are also options for policy year with calendar month, policy month with calendar year, and policy month with calendar month. The following table shows output for policy month with calendar year.

```
addExposures(records, type = "PMCY")
```

Policy Number	Policy Year	Policy Month	Start Interval	End Interval
Note: Table has been truncated				
B10251C8	1	8	2010-11-10	2010-12-09
B10251C8	1	9	2010-12-10	2010-12-31
B10251C8	1	9	2011-01-01	2011-01-09
B10251C8	1	10	2011-01-10	2011-02-09

Note: Table has been truncated

Lower_Year

The lower_year argument allows for left truncation by calendar year. Exposure rows will only be created if the interval contains no dates from years below the lower_year argument. This can reduce computation time and memory use.

```
addExposures(records, type="PY", lower_year=2017)
```

Policy Number	Policy Year	Start Interval	End Interval
B10251C8	8	2017-04-10	2018-04-09
B10251C8	9	2018-04-10	2019-04-04
D68554D5	13	2017-01-01	2017-12-31
D68554D5	14	2018-01-01	2018-12-31
D68554D5	15	2019-01-01	2019-04-04

Determine Output Size Before Calling addExposures() Using expSize()

The expSize function calculates an upper bound for the number of rows created by a call to addExposures but doesn't create the exposures. The expSize function runs faster and uses less memory than addExposures for large outputs so it can be useful.

```
expSize(records, type = "PY")
```

Upper Bound on Output Size
25

Joining Additional Information to Exposures

The call to addExposures removed the issue age and gender fields. We add these fields back by joining our original records to the created exposures using the policy number as the join criterion. In the next section, we discuss how to join by both a policy number and date.

Policy Number	Policy Year	Start Interval	End Interval	Issue Age	Gender
B10251C8	1	2010-04-10	2011-04-09	35	M
B10251C8	2	2011-04-10	2012-04-09	35	M
B10251C8	3	2012-04-10	2013-04-09	35	M

Note: Table has been truncated

PREMIUM PATTERN

Suppose we would like to analyze premium pattern by policy month for some transaction data.

```
trans
```

Policy Number	Transaction Date	Amount
B10251C8	2012-12-04	199
B10251C8	2013-12-28	197
B10251C8	2015-12-30	177

Note: Table has been truncated

We create monthly exposures called exposures_PM using the addExposures function. Later we join aggregated transaction data to these exposures.

```
exposures_PM <- addExposures(records, type = "PM")
exposures_PM
```

Policy Number	Policy Year	Policy Month	Start Interval	End Interval
B10251C8	1	1	2010-04-10	2010-05-09
B10251C8	1	2	2010-05-10	2010-06-09
B10251C8	1	3	2010-06-10	2010-07-09

Note: Table has been truncated

Allocating Transactions

The addStart function adds a start interval column to the transaction data that corresponds to the correct start interval from the exposure data frame. The start interval and policy number columns specify what row of the exposures_PM data frame a transaction should be allocated to.

```
trans_with_interval <- addStart(trans, exposures_PM)
trans_with_interval
```

Start Interval	Policy Number	Transaction Date	Amount
2010-05-10	B10251C8	2010-05-28	190
2010-06-10	B10251C8	2010-07-04	189
2010-11-10	B10251C8	2010-11-21	179

Note: Table has been truncated

We can group and aggregate transactions by policy number and issue date to get transaction totals for intervals in exposures_PM.

```
trans_to_join <- trans_with_interval %>%
  group_by(`Start Interval`, `Policy Number`) %>%
  summarise(`Total Amount` = sum(Amount))
trans_to_join
```

Start Interval	Policy Number	Total Amount
2005-06-01	D68554D5	97
2005-10-01	D68554D5	169
2005-12-01	D68554D5	96

Note: Table has been truncated

Then we can left join the aggregated transactions to the exposures data frame with join criteria of policy number and start interval.

Policy Number	Policy Year	Policy Month	Start Interval	End Interval	Total Amount
B10251C8	1	1	2010-04-10	2010-05-09	NA
B10251C8	1	2	2010-05-10	2010-06-09	190
B10251C8	1	3	2010-06-10	2010-07-09	189
B10251C8	1	4	2010-07-10	2010-08-09	NA

Note: Table has been truncated

We then fill in NA values with zero.

Policy Number	Policy Year	Policy Month	Start Interval	End Interval	Total Amount
B10251C8	1	1	2010-04-10	2010-05-09	0
B10251C8	1	2	2010-05-10	2010-06-09	190
B10251C8	1	3	2010-06-10	2010-07-09	189
B10251C8	1	4	2010-07-10	2010-08-09	0

Note: Table has been truncated

Now we are done with the data wrangling; from here it is not hard to calculate things like average premium per policy year or policy month. We can even export this data as a .csv to make dashboards in a business intelligence tool.

Other Uses for addStart()

Suppose we were interested in the last non-zero monthly premium paid by a policy. We left join the aggregated premiums to the exposure frame as we did before. This time we fill in NA values with the previous paid premium instead of 0. The first interval is NA because there are no prior premiums.

Policy Number	Policy Year	Policy Month	Start Interval	End Interval	Total Amount
B10251C8	1	1	2010-04-10	2010-05-09	NA
B10251C8	1	2	2010-05-10	2010-06-09	190
B10251C8	1	3	2010-06-10	2010-07-09	189
B10251C8	1	4	2010-07-10	2010-08-09	189
B10251C8	1	5	2010-08-10	2010-09-09	189
B10251C8	1	6	2010-09-10	2010-10-09	189

Note: Table has been truncated

Data manipulations like this can engineer features for anything varying with time like account values, guarantees or planned premiums.

DECREMENT RATES

Calculating mortality and lapse rates is not difficult once we have created the exposure data frame. In the following example, we calculate the exposure as

$$\frac{(\#\text{Days in Interval})}{365.25}.$$

It is not difficult to add a death indicator and use a full exposure in the year of death for performing a mortality study.

Policy Number	Policy Year	Start Interval	End Interval	Exposure	Death Count
Note: Table has been truncated					
D68554D5	12	2016-01-01	2016-12-31	1.002	0
D68554D5	13	2017-01-01	2017-12-31	0.9993	0
D68554D5	14	2018-01-01	2018-12-31	0.9993	0
D68554D5	15	2019-01-01	2019-04-04	1	1

We then aggregate by duration to calculate mortality rates.

```
exposures_mort %>%
  group_by(`Policy Year`) %>%
  summarise(q = sum(`Death Count`)/
  sum(`Exposure`))
```

Policy Year	q
Note: Table has been truncated	
8	0
9	0.5002
10	0
11	0
12	0
13	0
14	0
15	1

Expected Mortality

Some mortality tables have been loaded to the package in an easy-to-join format so that users can conduct an actual to expected analysis.

```
expstudies::mortality_tables
```

mortality_tables
AM92
GAM1983
UP1984
VBT2015_SmokerDistinct_ALB
CSO2017_Loaded_PreferredStructure_ALB

We view the AM92 Ultimate table.

```
mortality_tables$AM92$AM92_Ultimate
```

Attained Age	q Ultimate
19	0.000587
20	0.000582
21	0.000577
22	0.000572
23	0.000569

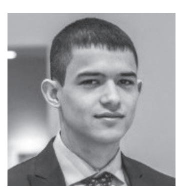
Note: Table has been truncated

We can join the mortality table to a data frame using the attained age as the join criterion for actual to expected analysis of calculated rates.

OPPORTUNITIES

It would not be difficult to implement the methods in this package in Python using pandas. In R, Python or Apache Spark, there is potential for running really large experience studies by parallelizing calculations. It would be nice if there was a framework for experience study calculations that has been reviewed by many people so that others are comfortable relying on the framework.

There is a question I am curious about the answer to. For a given data set and product specification, about would different organizations produce materially different models? I don't think there is much room for difference in lapse/mortality rate implementations, but there are many approaches that can be taken for something like premium pattern. Should we classify policies into separate premium schedules based on some combination of characteristics? Should we model future premiums as a percentage of planned premium? I think it would be interesting to have some sample data sets and people can produce and share simple universal life models in Excel or Python to compare different modeling practices. I don't think there is much research in the field that is fully reproducible due to data privacy concerns and proprietary modeling systems, so there is lots of work that can be done. In the future maybe actuarial organizations will sponsor things like open source economic scenario generators or open source models. ■



Matthew Caseres is an actuarial analyst at Ameritas. He can be reached at *matthew.caseres@ameritas.com* and on GitHub at *github.com/ActuarialAnalyst*.

SOCIETY OF ACTUARIES®

475 N. Martingale Road, Suite 600
Schaumburg, Illinois 60173
p: 847.706.3500 f: 847.706.3599
w: www.soa.org

NONPROFIT
ORGANIZATION
U.S. POSTAGE
PAID
SAINT JOSEPH, MI
PERMIT NO. 263