

Horses for Courses

By Van Beach

Complexity is a challenge in the insurance industry. Products, regulations, and the underlying risks of insurance are difficult to quantify, manage, and explain. Actuarial modeling has felt the tension created by complexity very keenly. Some examples are the following:

- Reserve and capital paradigms have shifted between formulaic and principle-based, vastly increasing the volume of calculations, data, and analyses.
- The range of applications for modeling has expanded tremendously over the last 10 years, putting increased strain on modeling systems.
- Actuarial model processes for data, assumptions, and reporting have become more complex and intensive.
- Relatively new concepts such as modeling efficiency approaches (e.g., cluster modeling) have become part of the modeling process.
- The infrastructure needed to support actuarial calculations has moved from (a) desktop processing to (b) on-premise grids to (c) cloud computing.
- Product designs and the associated methodologies and approaches for managing risks have diverged and become more proprietary as companies pursue competitive advantage.

Further, all of this needs to be governed. It is not an option, but a reality, for all companies. Modeling has changed and evolved rapidly, and it has left models and processes at many companies deficient. It is a complex challenge, and many companies struggle.

Companies intuitively know that change is required, but knowing what and how to change—even knowing the right questions to ask—is itself a challenge. In the attempt to address complexity, many of the debates have been reduced to oversimplified dichotomies:

- Open versus closed code,
- Single versus multiple systems and
- Desktop versus cloud platforms.

Of these, the open versus closed debate is the most longstanding and has the most fundamental impact on the complexity issues noted above. This article will explore the context of the debate, along with the pros and cons of each approach, and will conclude with a viewpoint on the “right” approach.

WHAT IS THE OPEN VERSUS CLOSED DEBATE IN THE CONTEXT OF ACTUARIAL MODELING?

In an actuarial modeling context, the open versus closed debate refers to the actuarial code required to support a model:

- An open code approach allows the user to view and modify the calculation of an actuarial model directly by adding, deleting, or changing business rules. The vendor typically provides standard code, but the user can augment the standard code with proprietary logic.
- The user cannot view or change the business rules under a closed code approach. Instead, the code is maintained by the software vendor. The user can change only the input parameters to the system. Code is reviewed indirectly through examples or documentation.

The open versus closed debate is not limited to actuarial modeling. It is pervasive in software engineering, and proponents are almost fanatical in their defense of both approaches.

WHERE HAS THE INDUSTRY BEEN ON THIS ISSUE?

Actuarial modeling found its roots in a closed code environment on mainframe computers. The introduction of desktop PCs changed the game, when the power to create and innovate was shifted to the end user. New software vendors and actuarial modeling products entered the market, and options now included the following:

- Vendor systems with entirely closed actuarial logic,
- Closed vendor systems with insertion points or formula tables to enable customized logic to augment the core,
- For the savvy modeler with programming skills, models that could be built from scratch using powerful desktop programming language packages—the truly open code solution and
- Systems built with closed frameworks around flexible scripting languages providing a blend of open and closed code.

The modeling market evolved, providing options across the full range from entirely closed to entirely open. As market needs evolved—driven by product innovations, new risk management approaches, and new regulations—and companies gained experience with the pros and cons of open and closed systems, companies’ preferences have evolved and shifted as well. At some points the scale tilted toward open approaches, at other times favoring closed approaches.

WHAT ARE THE PROS AND CONS? ACTUARIAL MODELING CONTEXT

As noted above, industry preferences have evolved as market requirements have evolved, but the debate is still ongoing. The pros and cons of open and closed approaches to actuarial modeling are not as simple as writing a list. Understanding the benefits and disadvantages of open versus closed approaches



first requires an exploration of several key interdependent concepts, including the following:

- The universe of actuarial modeling calculations,
- Actuarial modeling applications and the need for flexibility and control,
- Required level of precision and
- Continuous change in products, regulations, risks, and modeling approaches.

Each of these concepts will impact key considerations such as speed, ease of use, scalability, quality, and cost, and they are discussed below.

The Universe of Actuarial Modeling Calculations

Actuarial model calculations are substantial because they should encompass all material product, asset, company, economic, regulatory, and risk characteristics. They may include the following:

- Lives in-force, reflecting interdependent decrements such as mortality, morbidity, voluntary surrenders, and lapses;
- Product features and policy mechanics (e.g., account value crediting, dividend payments, or mode of care benefit maximums);
- Guarantees such as minimum cash values, death benefits, or withdrawal values;
- Commissions, expenses, and other company cash flow items
- Asset characteristics;
- Reserve regulations, both formulaic and principle-based;
- Capital requirements, both formulaic and principle-based;
- Investment strategies;
- Future business issued;
- Economic impacts, including policyholder behavior and interest-crediting methodologies;
- Accounting structures;
- Taxes;

- Distribution of profits and
- Management behavior and reactions.

Although the general categories of calculations are common, it is critical to understand the heterogeneous nature of the logic; that is, two companies that issue the same products to the same market in the same jurisdiction could require substantially different logic to account for company-specific requirements across any one of the categories noted above. Further, considering a global market where regulations and products differ across nearly every jurisdiction, the universe of actuarial logic expands even more. Good design should reuse common components and calculations, but even when optimized, the full breadth of calculations is staggering.

But while the universe of calculations is staggering, what is required for a given company for a given application is just a subset. From a systems standpoint, a specific application for a given company needs only the applicable logic. For the focused purpose, the other options and features are just clutter. So by providing a comprehensive solution to a wider range of clients, the system becomes more complex for all users. Thus, there is an inherent tension between comprehensiveness and tractability.

Actuarial Modeling Applications, Flexibility, and Control

Actuarial models can be broadly classified into three categories:

- Pricing,
- Valuation and
- Projections.

Although the core aspects of the model may be the same, the processes and control requirements around these three functions differ greatly.

When working with a model in a pricing context, flexibility is important to test different product designs, benefit structures, risk management approaches, and the like. The ability to explore, understand, and creatively adjust nearly all aspects of the actuarial model is desirable.

With a valuation model, changes are made with much more control. Examples include projections to support FAS 97 GAAP, Principle-Based Reserves, Solvency II, IFRS, and other model-based regulations. Hedging analysis would also have characteristics similar to valuation with regard to the need for high levels of control and productionization. For these applications, each model change has the potential to impact a reported financial result or a critical financial measure, and therefore these applications have a very low tolerance for mistakes. In this environment, control and governance are key.

The middle ground of “projection” applications can show aspects of each. Cash flow testing, ERM and economic capital projections, duration matching, and many other applications are generating results under increasingly rigorous controls, yet there is typically more iteration and “what-if” analyses than for valuation applications.

In short, the requirements for the style of usage of models differs significantly across applications, ranging from flexible to highly controlled.

Required Level of Precision versus Speed

By definition, a model does not produce a “correct” result. A model is a representation of reality, used to analyze potential future outcomes. The level of detail necessary in a model to capture the relevant characteristics of the vast array of products, regulations, risk management approaches, modeling methodologies, and other things again varies by many factors and is not heterogeneous.

The granularity and detail of calculations also has a direct relationship with the time required to complete a model projection. As the precision of the model increases, so does the time required for the model to execute. Model runtime is a critical factor impacting the usefulness of a model, so finding the right balance of precision and speed is another area of inherent tension in model design.

Continuous Change in Products, Regulations, Risks, and Modeling Approaches

Development of a model is never finished. New products, new regulations, and new modeling approaches produce evolving and changing models. Some changes, such as new products, are driven by internal demands and will reflect company preferences. Other changes, such as new regulations, are external and often subject to interpretation, especially as the regulation intersects with company-specific designs. Companies must address how model changes will be implemented and managed. The design, process, and timing of the implementation are as often critical as the change itself.

WHAT ARE THE PROS AND CONS?

With the context in mind, let’s look at some of the pros and cons of each approach:

Closed: Pros

- The vendor likely has experts and specialized expertise to develop, implement, and optimize the functionality it provides. With complete control and knowledge of the system, it can likely implement a given feature faster and with better quality than a modeler.
- The code is also the responsibility of the vendor, so the expertise to develop and maintain the code does not need to be maintained by the modeler.

- The code is common for all modelers, so there is the potential for greater review and more efficient system support.
- Code consistency is ensured because all the code is guaranteed to be the same.
- The code can be optimized by the vendor and, if done well, will not impact the model results.
- Every modeler gets the benefit of new logic introduced through system upgrades.
- System upgrades, including logic, can be more seamless and streamlined.
- The modeler can rely on the vendor to provide the code they need when they need it.

The binary open versus closed debate greatly oversimplifies the reality of actuarial modeling options.

Closed: Cons

- The closed system may simply not be able to do what a company needs it to.
- Every modeler gets the weight and complexity of new logic and features, making the model increasingly intractable.
- As the system grows to accommodate more features and functions, the complexity increases.
- Similarly, runtime will likely degrade as the system grows.
- The timeline for new features, options, methodologies, and the like is at the discretion of the vendor. The modeler is taking on the business risk that the vendor will provide the code they need when they need it.
- Proprietary products and methods may be exposed to others once implemented.
- Debugging is more challenging since the code can be analyzed only indirectly.
- Creativity is limited since only data can be changed.

Open: Pros

- The system can be optimized for the modeler’s needs. Features can be added, changed, hidden, or removed.
- Data and code complexity can be greatly reduced by introducing targeted changes.
- The modeler can optimize the performance of the model.
- Code can be written that aligns perfectly with the company’s view of products, regulations, and risk management, improving tractability.
- Proprietary code is guaranteed to remain proprietary.
- Debugging and understanding calculations is more rapid since the code can be viewed directly.

- The modeler can make changes according to his or her timeline.
- Flexibility exists to lock down (essentially, “close”) elements of the calculations once they have been finalized.

Open: Cons

- Poorly implemented code can result in poor performance, incorrect results, or unnecessary complexity.
- Expertise in managing code needs to exist within the company or be purchased from outside sources.
- Knowledge of the code must be maintained within the company.
- Code needs to be maintained, and changes must be documented and governed.
- If coding changes are allowed to proliferate across a company, the effort required to maintain consistency within the company will grow, and reconciliation will become increasingly challenging.
- Upgrading the logic is typically not seamless and requires extra effort to implement and test.
- Standard logic provided by the vendor will typically be limited to common features and approaches, not the entire universe of calculations.

IS THERE A RIGHT ANSWER?

Exploring the “right” answer requires that we go back to the discussion of concepts that frame the open versus closed debate. First, the reality of the breadth of actuarial calculations is an important consideration. A closed system will take on that entire burden. An open system will leave some of this burden with the modeler. At what point does the closed system reach a tipping point where the sheer volume of code and calculations becomes overly burdensome? Or does the closed approach naturally lead to systems that are targeted for more specific applications or jurisdictions where the calculations can be tailored and focused (hedging, for example)? Can an open system hit the sweet spot of providing core calculations and leaving only the truly custom components to the modeler? Will an open system fall victim to the temptation of trying to provide too many features? Or too few features leaving too much customization for the client to meet all their needs?

Also, code is only part of the challenge. Data are equally as difficult to manage, and data errors are just as costly as code errors. Data requirements are necessarily greater with closed systems. This is especially true for closed systems as they become more comprehensive. With closed code, additional parameters and inputs are necessary to provide the required options and functionality. Often with an open system, a simple line of code can eliminate the need for multiple tables, inputs, and other data. As discussed earlier, the universe of potential actuarial calculations is staggering. Data can be staggering as well. Both must



be considered, and the best answer may be to find a balance between both, meaning that flexible code is part of the solution to reducing data complexity.

What about quality of code and maintenance burden? A closed system addresses that challenge and limits proliferation of code by locking it away. That is one solution—albeit an extreme one. It is sometimes viewed positively because companies have felt the pain from poorly managed open environments. This attitude is more common among companies that have been using open systems for the last decade during which modeling was an ad hoc exercise and products and modeling approaches were still evolving. Over time, model governance best practices emerged (partly in response to the inherent provided creativity), and many companies incorporated vendor standard code; however, not everyone took the time for this exercise. The lesson is that power and flexibility must be actively managed.

Change is a given, so models, data, and code need to evolve. Regardless of whether a system is open or closed, the model and data need to be governed. As discussed above, data governance is equally as critical and complex as code governance. With regard to code, a closed system again will be solely responsible to address this requirement, often with direction from clients’ companies. For many this is a significant benefit because code development is a specialized skill that companies do not want to maintain. However, once a company does, in fact, recognize that model development, including both code and data, is a specialized skill and does not expect all their modelers to be developers, the landscape changes. Centralizing and optimizing model development and change is a key evolution in how companies organize their modeling function. With limited and controlled access to the code, data, and configurations, changes can be governed effectively. With an open model, clients have an option to develop and innovate logic at their own pace, using vendor-provided code as appropriate and available. As noted above, vendor-provided code will take more effort to incorporate into

an open system, but with a centralized modeling function, that burden is greatly reduced.

An open system can be closed. A closed system cannot be opened. A closed system provides a narrow range of flexibility at the closed end of the spectrum. An open system provides a range of options from closed to open, giving responsibility to manage that flexibility to the user. It is quite likely that the right answer is to not choose a single point on the spectrum, but rather use the appropriate approach for the given company and application, which is an option available to open systems but not available to closed systems.

CONCLUSION

Each system has an approach for meeting the needs of its modeling customers. Some companies preach the benefits of closed code. Others promote the benefits of flexible, open code.

The binary open versus closed debate greatly oversimplifies the reality of actuarial modeling options. Nearly all vendor systems reflect a blend of open and closed components:

- Systems purporting to be entirely closed offer formula tables and insertion points.
- Systems that are perceived as open are built with underlying frameworks and architectures that lock away certain fundamental modeling calculations and have varying levels of access to modify calculations.

Each system reflects its preferred approach to blending open and closed capabilities. As tools evolve to better manage and

govern code, open systems have the potential to be governed with confidence and assume many of the positive aspects of closed systems. Because of the various blends of features, each system will need to be evaluated with a keen eye toward understanding how flexibility and control are provide via open and closed aspects of their system approach. Remember, though, that an open system can be closed. A closed system cannot be opened.

There is no “right” answer for every company and every situation. The discussion of the context for actuarial modeling is critical—these realities directly impact whether the flexibility of an open system or the control of a closed system is the best choice for a given company. But it is interesting again to observe that even within a company, different functions likely prefer vastly different approaches (i.e., pricing prefers open versus valuation that prefers closed).

So choose the horse for your particular course. Or choose to have a stable of horses all trained to excel at each and every race that is important to you.

If this article sounds familiar, it is because an article with the same title was written by Phil Gold in the January 2007 *CompAct*. Many of the discussions from 2007 are revisited here. ■



Van Beach, FSA, MAAA, leads the professional services organization within the Life Technology Solutions (LTS) practice for Milliman. He can be contacted at van.beach@milliman.com.