Designing Quality into Expert Systems: A Case Study in Automated Insurance Underwriting

Kareem S. Aggour, Janet A. Barnett, Piero P. Bonissone

GE Global Research
One Research Circle
Niskayuna, NY 12309, USA
aggour@research.ge.com, barnettja@research.ge.com, bonissone@research.ge.com

Abstract

It can be difficult to design and develop artificial intelligence systems to meet specific quality standards. Often, AI systems are designed to be "as good as possible" rather than meeting particular targets. Using the Design for Six Sigma quality methodology, an automated insurance underwriting expert system was designed, developed, and fielded. Using this methodology resulted in meeting the high quality expectations required for deployment.

Introduction

Six Sigma

Six Sigma originated as a methodology for applying statistical rigor to improve manufacturing processes. The methodology was developed to pinpoint sources of defects, identify their root causes, and assist engineers in searching the design space for solutions. The term "Six Sigma" describes the goal of reducing process variability to the point where the short-term mean is at least six standard deviations (sigmas) away from any control limits on the output. This results in no more than 3.4 defects per million opportunities in the process in the long term.

GE has been utilizing the Six Sigma methodology for over seven years to improve manufacturing processes. While these techniques proved valuable for reducing defects in existing processes, they did not fully address new product design and development. Therefore, these techniques were adapted to the design of new systems; this adaptation, called Design for Six Sigma (DFSS), involves designing quality into a process or system. By using Six Sigma rigor during system design and development, most quality issues during production can be avoided altogether.

The DFSS process is composed of several steps represented by the acronym DMADOV (Hoerl 2001):

- Define: determine the scope of the problem to be solved; identify CTQs (Critical To Quality system characteristics)
- Measure: validate the measurement system used for determining the system quality
- Analyze: identify approaches to solve the problem and perform trade-off studies
- Design: perform system design; evaluate overall system quality based on the design
- Optimize: utilize statistical rigor to optimize the design
- Verify: confirm the system meets the requirements; define processes to ensure that the system will continue to meet the CTQs over time

The DFSS methodology can be applied to the design and development of a new AI system as readily as it can be applied to the design of a new refrigerator or gas turbine. This paper describes how the DFSS steps were used to develop an expert system that classifies insurance applicants into discrete risk categories. The following section introduces the insurance underwriting problem. The remaining sections describe how each of the DFSS steps was applied to solve the problem.

Medical Insurance Underwriting

Traditionally, a human underwrites GE medical insurance applications. These 'underwriters' are responsible for reviewing applications and, based on the applicant's medical history, assigning the applicant to a discrete risk category that dictates the premium to be paid for the insurance (or declining the applicant altogether). The higher risk category, the higher the premium.

Underwriters follow business guidelines specified in an underwriter manual, but also rely upon extensive medical knowledge and personal experience when underwriting cases. The fact that they use their own experience and judgment to make decisions makes this a difficult problem to automate. It is critical to the business that applicants are placed in appropriate risk categories. Underestimate the risk category and the applicant would not pay enough to cover the financial risk GE incurs by insuring the individual. Overestimate the risk category and GE insurance will not be price competitive; they will be unlikely to win new or retain existing customers.

Automating this process has a number of benefits, including improving consistency and reducing the number of defects while allowing the case volume to grow. Reducing defects allows GE to remain price competitive while effectively managing risk. In addition, if the automated process can efficiently generate decisions on a large volume of cases, the capacity of the underwriting process will also improve.

Previous work describes the development of an automated underwriting system to solve GE's 'clean' (unimpaired) cases (Aggour & Pavese 2003). The present problem addresses automating the underwriting of more complex cases: applicants who suffer from hypertension. This requires the development of a discrete AI classification algorithm with three placement categories.

Define

The main step of the DFSS Define phase is to identify the project goals in terms of CTQs. As stated previously, CTQs are aspects of the algorithm that are deemed Critical To Quality. The customer typically defines the CTQs. They should be both quantifiable and measurable. For this problem, the customer identified two requirements for automating their insurance underwriting processes:

1. Accuracy no worse than the current human-based underwriting process
2. Touch-free operation

While these two requirements were very important, they were not Six Sigma CTQs (they were neither quantifiable nor measurable). After additional discussion, these responses were refined. The final CTQs were:

1. The algorithm must produce decisions that agree with the business underwriting staff. The rate of agreement must equal or exceed that of current manual assessments.
2. Coverage of 50% of insurance applications (assuming historical application distribution).

Note that "touch-free operation" was determined to be unrealistic since even human underwriting sometimes requires validation by experts. Additionally, it would be impossible to develop an automated system in anticipation of all possible scenarios. Additional CTQs were identified, but they are beyond the scope of this paper.

Measure

Two things are needed to validate the underwriting algorithm: (1) a set of solved cases against which the algorithm can be validated, and (2) a clear definition (or gauge) of how to measure for rate misclassifications.

Gold Standard Cases

A set of accurate decisions was needed before the automated decision algorithm could be verified. These decisions, a standard reference data set of "gold standard" cases, were taken from a stratified random sample of the historical population of applicants suffering from hypertension and no other impairments. Expert underwriters then verified the decisions. Comparing the original underwriter decisions on these cases to the expert decisions produced a benchmark of the current process. Cases were filtered out of the gold standard data set that contained incomplete data or were outside the scope of the algorithm (e.g., cases with impairments other than hypertension, as well as cases with multiple impairments).

Additional validation of gold standard cases was performed via consistency checking. Attributes of the cases were defined such that "less is better" and therefore, a partial ordering of the cases could be performed. In this manner, it was possible to identify incorrect rate classes based on a dominance relation. If two cases, A and B, have values such that for all inputs, $X_A \leq X_B$, then their corresponding rate classes must also reflect the same ordering, i.e., $Y_A \leq Y_B$. Case consistency was improved by eliminating cases that were incompatible with this dominance relation (Figure 1).

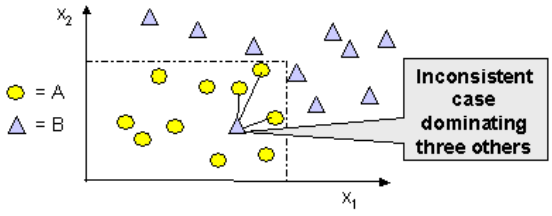

Figure 1: Case Consistency for Two Attribute Example

Measurement System (The Gauge)

In this example, a measurement system must be defined that will accurately characterize how well the algorithm is performing compared to the expert human underwriters. Any decision that the algorithm makes will be either correct or incorrect according to the underwriters; however, because there are multiple, ordered rate classifications, more information than correct/incorrect can be captured. Type I/Type II error classification is insufficient for capturing the accuracy of the algorithm.

A confusion matrix, a popular measurement tool for pattern recognition, allows for measurement of the relative distance among results. A matrix is used to correlate algorithm results with the known outcomes (Figure 2). Counts of results are placed in each cell; for example, if 200 cases were correctly categorized as A, then "200" would appear in the upper-leftmost cell. Entries along the

main diagonal are correct. Entries in all other cells are "confused" for other outputs (they represent incorrect classifications). For ordered outputs such as these, entries that are further away from the main diagonal represent larger errors in classification.

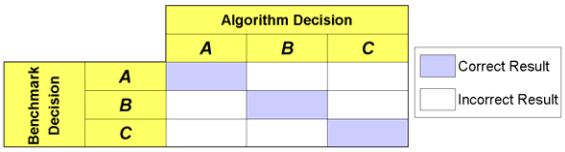

Figure 2: Confusion Matrix Structure

This measurement system relies upon the accuracy of the expert underwriters' decisions for comparison, both for initial validation as well as for periodic checks after the system is put into production. While expert underwriters will be used for algorithm validation, this still presents challenges: experts may disagree on rate classifications and even make mistakes. This is not a perfect measurement system, but it is the best available. Because of these imperfections in the gauge, it is important not to take the expert decisions at face value but to ensure that they are consistent with previous decisions.

Analyze

During the Analyze phase of DFSS, the design space is explored for solutions to meet the CTQs. This process can include: comparing design alternatives, developing designs to meet reliability requirements, and creating a scorecard for measuring the quality of the algorithm throughout the life of the system.

Trade-Off Analysis

A design trade-off study was performed to select an approach for the automated underwriting algorithm. Based on the design team's past experiences, four alternatives were selected for the study: neural nets, fuzzy logic rules, case-based reasoning, and multivariate adaptive regression splines. Evaluation criteria were developed based on the CTQs described earlier. For example, accuracy, consistency, coverage, and ease of use were a few of the many criteria evaluated for each alternative. For each alternative, each criterion was assigned a value from 1 (low applicability) to 5 (high). The result of this analysis clearly indicated that fuzzy logic rules were most applicable to this application. In particular, the fuzzy logic rules surpassed the other alternatives in consistency and ease of optimization.

Design for Reliability

It is important not only to ensure that the algorithm performs correctly initially, but also to ensure that the algorithm performs well over time in the face of change. In this insurance underwriting example, change is inevitable. The characteristics of the applicant pool change over time, the business' underwriting rules change, and government regulations change; each may require updates to the automated underwriting algorithm.

One of the keys to long-term reliability of any algorithm is to design for extensibility. This means using a component methodology that enables developers to easily make changes. It also means making sure algorithm and software system configuration parameters are not hard-coded, but are accessible in configuration files.

Defining a Scorecard

A Six Sigma scorecard is a mechanism for tracking the quality of the algorithm's decisions. The confusion matrix described earlier can be thought of as the measuring stick, while the scorecard is the report card. The scorecard provides an instant view into how well the system is performing.

To create a scorecard, a set of metrics must be defined that will characterize the quality of the system. For the automated underwriting algorithm, three metrics were selected: coverage, relative accuracy, and global accuracy. These metrics are defined in Figure 3.

Coverage: Total number of decisions made as a fraction of the total number of cases (N)

Coverage = $\Sigma\Sigma\, M(i, j)/N$

Relative Accuracy: Total number of correct decisions as a fraction of the total number of decisions

Relative Accuracy = $\Sigma M(i,i)/\Sigma\Sigma\, M(i,j)$

Global Accuracy: Total number of correct decisions as a fraction of the total number of cases

Global Accuracy = $\Sigma M(i,i)/N$

Figure 3: Definitions of Scorecard Metrics

These metrics can be evaluated from the data contained within the confusion matrix. A confusion matrix and scorecard containing the performance of the human underwriters is shown in Figure 4.

		Human Underwriter Decision		
		A	B	C
Benchmark Decision	A	1	1	0
	B	1	269	43
	C	0	39	46

Coverage	100%
Relative Accuracy	79%
Global Accuracy	79%

Figure 4: Human Underwriter Confusion Matrix and Scorecard

Design

The objective of the Design phase is to develop the transfer function and evaluate its initial effectiveness. In Six Sigma terms, the transfer function quantitatively describes the relationship between the critical inputs (X's) and the output (Y). In DFSS for algorithms, the transfer function can be thought of in terms of the inputs and outputs of the actual algorithm. The parameters to an algorithm transfer function are the variables that affect the behavior of the algorithm. In essence, the transfer function is equivalent to the core decision algorithm. As such, it should be implemented in such a way that all parameters are easily modified, and any other reasonable adaptations can be made without rewriting code. This allows the algorithm's parameters to be optimized in an efficient manner.

From the results of the design trade-off analysis (Analyze phase), it was decided to develop an algorithm based on fuzzy logic rules. Fuzzy logic is a superset of conventional Boolean (true/false or 1/0) logic, allowing values to be equal to any real number in the interval [0,1]. Intermediate values denote a "partial degree of satisfaction" of some statement or condition (Zadeh 1965). A Fuzzy Logic Rules Engine (FLRE) was designed and developed to implement the automated insurance underwriting transfer function. This transfer function takes as input a set of continuous X's (the applicant's medical information), and outputs a discrete Y—a risk classification for the applicant (Bonissone, Subbu, and Aggour 2002).

The FLRE encodes the underwriter guidelines into a set of fuzzy rules. These rules identify fuzzy cut-offs for each X being evaluated (for example, cholesterol levels) to determine the customer's risk placement. The objective of the FLRE is to identify the most competitive rate class for the applicant while ensuring that the combined effects of the applicant's medical factors meet the constraints imposed for that rate class. If a constraint is not satisfied for a rate class, then the applicant should be placed in the next best rate class. Refer to Aggour & Pavese (2003) for details on how the FLRE makes decisions.

After implementation of the FLRE algorithm, the baseline performance of the algorithm was measured, producing the initial results found in Figure 5.

		Fuzzy Logic Rules Decision		
		A	B	C
Benchmark Decision	A	2	0	0
	B	1	308	4
	C	0	5	80
Coverage				100%
Relative Accuracy				97.5%
Global Accuracy				97.5%

Figure 5: Initial Baseline FLRE Confusion Matrix and Scorecard

While the initial fuzzy cut-offs were determined from the underwriter guidelines (and sessions with the medical staff and actuaries), it was unknown whether these cut-offs were optimal.

Optimize

During the DFSS Optimize phase, the transfer function parameters are tuned to achieve optimal results from the algorithm. A tolerance analysis is performed to ascertain acceptable limits on the transfer function inputs to produce outputs that are within specification.

Once the baseline performance of the algorithm had been ascertained (Figure 5), the next step was to optimize the algorithm to improve its performance. An Evolutionary Algorithm (EA) approach was selected to automate the process of identifying optimal parameters for the algorithm (Goldberg 1989).

The FLRE uses an EA of chromosomes containing elements that represent each tunable parameter within the configuration. Since a chromosome defines a complete FLRE configuration, a new instance of the FLRE can be initialized and evaluated for each chromosome. Figure 6 shows the interaction of the EA (on the left) with the FLRE (on the right). For each generation of the EA, a separate

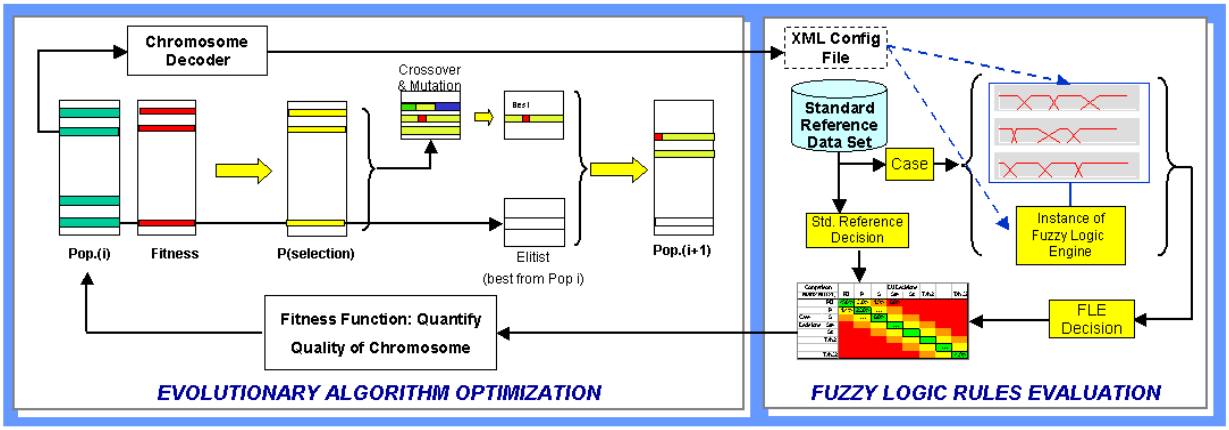

Figure 6: FLRE Optimization using Evolutionary Algorithms

instance of the FLRE was instantiated per chromosome (via the 'Chromosome Decoder' in Figure 6). The gold standard reference set of test cases was then used to evaluate the FLRE to determine how effective the parameter configuration was at producing the correct results (those results most closely matching that of the expert underwriter panel). From the confusion matrix, the EA fitness function assigned a fitness score to the chromosome, which was then used to determine if the chromosome would be discarded, selected to undergo mutation and/or be passed unchanged into the next generation.

From the confusion matrix in Figure 2, it is clear that an optimal set of parameters would generate a completely diagonal matrix. Therefore, the fitness function drives the EA to find a set of parameters that can produce this result, or come as close as possible to doing so. Once the EA optimization was complete, a configuration was found such that the final confusion matrix was in fact completely diagonal. The post-optimization confusion matrix and scorecard can be found in Figure 7.

		Fuzzy Logic Rules Decision		
		A	*B*	*C*
Benchmark Decision	*A*	2	0	0
	B	0	313	0
	C	0	0	85
Coverage				100%
Relative Accuracy				100%
Global Accuracy				100%

Figure 7: Optimized FLRE Confusion Matrix and Scorecard

Tolerance Analysis

In typical Six Sigma manufacturing process design, the next step would be to perform statistical tolerancing to determine acceptable limits on the inputs to produce parts within acceptable limits at the output. For this algorithm, however, there are predefined limits for each of the inputs, and so a standard tolerance analysis is not necessary. But it is important to understand the potential significance that variations in the inputs will have on the output. Since it is unknown if a drift in the input is actually affecting the output, limits on the variability of the outputs must be identified to determine when a drift in the output is occurring. Control charts are used to monitor variance in the output over time and to alert stakeholders when unexpected variance occurs that may result in defects. The key is to determine how much variability in the inputs can be handled before the algorithm begins to break down.

For this project, control chart limits that quantify the meaning of 'too much variance' in the outputs are unknown. Therefore, appropriate limits are dynamically determined by running the algorithm for a period of time. Once sufficient data has been collected on the execution of the algorithm, statistical analysis of the data enables the inference of appropriate control limits on the outputs. To verify these control limits, an additional pilot period was executed during which the control charts were monitored and the quality of the limits verified. The control charts will be discussed further below.

Verify

During the final phase in the DFSS process, the transfer functions are verified and the algorithm is transitioned to the customer. Verification involves confirming the customer CTQs have been met. A monitor and control plan is developed to watch key leading indicators of risk and identify a plan to mitigate any risks that appear. Finally, documentation on the whole system is completed, and training and transition is performed. Once the algorithm has been optimized, it must be verified that the parameters have not been over-optimized to the training data set. Therefore, an out-of-sample verification is performed (out-of-sample meaning test cases not previously seen by the algorithm were used). The results of the out-of-sample verification can be found in Figure 8.

		Fuzzy Logic Rules Decision		
		A	*B*	*C*
Benchmark Decision	*A*	3	0	0
	B	0	160	0
	C	0	0	37
Coverage				100%
Relative Accuracy				100%
Global Accuracy				100%

Figure 8: Out-of-Sample Verification Confusion Matrix and Scorecard

Confirm CTQs Satisfied

The out-of-sample verification is used to confirm that the CTQs have been satisfied. The customer CTQs were:

1. The algorithm must produce decisions that agree with the business underwriting staff. The rate of agreement must equal or exceed that of current manual assessments.
2. Coverage of 50% of insurance applications (assuming historical application distribution).

For the first CTQ, Figure 8 indicates the algorithm has a relative accuracy of 100%, versus the benchmark underwriter performance of 79% in Figure 4. This CTQ has been satisfied. For the second CTQ, Figure 8 also indicates that the coverage of the algorithm is 100%, so all of the cases sent to the engine were placed. Therefore, the second CTQ was also satisfied.

Monitor and Control Plan

The next objective is to develop a monitor and control plan for when the algorithm is in production. The objective of monitoring is to signal stakeholders when statistically significant changes to leading indicators occur. The objective of control is to define a clear process for stakeholders to follow when diagnosing the cause of signaled changes and determining the appropriate action plan. Through monitoring and control, business risk and profit loss should be minimized via proactively identifying and reacting to significant changes in the system and its environment.

A decision must be made as to which leading indicators of risk to monitor. Both inputs (X's) to the algorithm and its outputs (Y's) may be monitored. In this algorithm, the key output to monitor is the engine rate class decision distribution. The engine should produce a fairly consistent distribution of decisions from week to week. Slight variations are to be expected, but barring any significant changes in the inputs the distribution should be consistent. Additionally, the percentage of cases the engine sends to the underwriter will be monitored to ensure adequate coverage by the engine. Figure 9 shows a control chart tracking the percentage of cases sent to the human underwriter. The last two readings on the chart exceed the Upper Control Limit (UCL), and would cause the monitor to generate a notification.

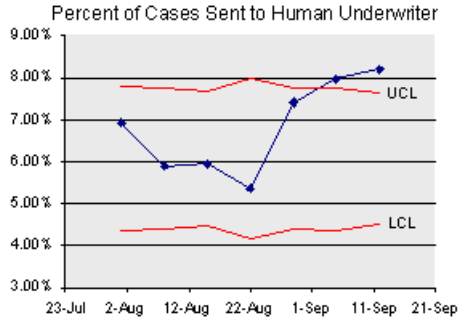

Figure 9: Control Chart Tracking

Document and Transition

The maintainers of the algorithm are often different from the maintainers of the software, so adequate documentation must be produced to satisfy the requirements of both teams. The algorithm maintenance team must have sufficient documentation to update and re-optimize the FLRE. The system maintenance team must have information on how to use, add to, or modify the software, and also how to deploy the application in various environments.

Writing thorough and comprehensible documentation is necessary but not sufficient. Knowledge transfer and transition ensures that the recipients of the deliverables, both from an algorithm and software system perspective, are trained and capable of using and maintaining the system. For this process, the team developed a set of training material on how to work with the algorithm, including how to modify rules and optimize parameters. This included:

- Defining a training curriculum customized for key personnel
- Defining and documenting step-by-step processes for making various common changes
- Delivering the training course at the customer site
- Reviewing hands-on examples to ensure personnel can make changes with assistance from the team
- Assigning "homework" to key personnel to ensure they can make changes without assistance

Conclusions

Using an established methodology such as Design for Six Sigma provided a valuable framework in which to design and develop an automated underwriting expert system. This system is responsible for automating the underwriting of medical insurance applicants suffering from hypertension. Using Six Sigma greatly facilitated the design, implementation, and transition process. It resulted in clearly defined goals that were met by the algorithm, enabling the successful fielding of the application.

Acknowledgements

We would like to extend our gratitude to Richard Messmer, Marc Pavese, Angela Patterson-Neff, Raj Subbu, and Kunter Akbay for their contributions to this work. We would also like to thank Martha Gardner for her input.

References

Aggour, K. and M. Pavese, 2003. "ROADS: A Reusable, Optimizable Architecture for Decision Systems", *Proceedings of the Software Engineering and Knowledge Engineering Conference*, San Jose, California, USA, pp 297-305.

Bonissone, P., R. Subbu and K. Aggour, 2002. "Evolutionary Optimization of Fuzzy Decision Systems for Automated Insurance Underwriting", *Proceedings of the IEEE International Conference on Fuzzy Systems*, Honolulu, Hawaii, USA, pp 1003-1008.

Engelmore, R. and A. Morgan, eds. 1986. *Blackboard Systems*. Reading, Massachusetts: Addison-Wesley.

Goldberg, D.E., 1989. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Reading, Massachusetts: Addison-Wesley.

Hoerl, R., 2001. "Six Sigma Black Belts: What Do They Need to Know," Journal of Quality Technology, vol 33, no 4.

Zadeh. L.A., 1965, "Fuzzy Sets," *Information and Control*, vol 8, pp 338-353.