ISSUE 35 | APRIL 2010

SOCIETY OF ACTUARIES Technology Section

CompAct

ELECTRONIC NEWSLETTER

CONTENTS

QUICK LINKS

Technology Section Web site

Council

Links of Interest

Fiction Contest

Howard Callif, Editor

SOA Staff
Meg Weber, Staff Partner

Sue Martz,

Share Print Article Search Back issues

LETTER FROM THE CHAIR

by Tim Deitz

When my daughter, who is now a sophomore in college, was in high school she wanted text messaging on her phone. My immediate answer was "No." It costs more money and the only purpose I saw was to communicate when you weren't supposed to, like during class! Well, she worked on my wife and negotiated a deal.

[full article]

EDITORIAL

by Howard Callif

People have submitted feedback on our new online format. Much of it is positive, but there is a group of people that want the newsletter in a printed format. We are looking into adding functionality to the Web site so all of the articles can be printed, so everyone can view the newsletter in the format that is convenient for them.

[full article]

HORSES FOR COURSES

by Phil Gold

We live in an environment where the word open has positive connotations, while closed has a negative feeling to it. Linux is open source, and Windows is proprietary or closed. So Linux must be better, right? Well, for some people it is, and if you read the Internet blogs, there's no contest.

[full article]

Section Specialist

Sam Phillips, Staff Editor

VISUAL BASIC NET REFERENCES

by Howard Callif

This article will provide useful references for learning and using Visual Basic NET.

[full article]

R CORNER-GRAPHICS

by Steve Craighead

The graphical capabilities of R are extensive. R is truly an object-oriented language with classes built upon classes.

[full article]

NUMBER PUZZLE

by Marc Whinston

Test your problem solving skills with this fun puzzle.

[full article]

ARTICLES NEEDED

by CompAct

Your help and participation are needed: Articles, book reviews and other information on technology useful to actuaries are needed!

[full article]

Actuaries
Risk is Opportunity.®

475 North Martingale Road, Suite 600 Schaumburg, Illinois 60173
Phone: 847.706.3500 Fax: 847.706.3599 www.soa.org

ISSUE 35 | APRIL 2010

SOCIETY OF ACTUARIES Technology Section

CompAct

ELECTRONIC NEWSLETTER

QUICK LINKS

Technology Section Web site

Council

Links of Interest

Fiction Contest

Howard Callif, Editor

SOA Staff
Meg Weber, Staff Partner

Sue Martz,

Share Print Article Search Back issues

LETTER FROM THE CHAIR

by Timothy Deitz

When my daughter, who is now a sophomore in college, was in high school she wanted text messaging on her phone. My immediate answer was "No." It costs more money and the only purpose I saw was to communicate when you weren't supposed to, like during class! Well, she worked on my wife and negotiated a deal. She had a B in her math class. If she pulled it up to an A, then she could get text messaging. Of course, she got an A, so I was on the hook. She had a limited number of texts per month and we monitored it closely. She had to pay for any overage charges.

I was talking to my cellular provider one day regarding a different issue and they said I could get unlimited texting on all five of my phones for just 10 dollars a month more than I was paying for the limited number on the one phone. I saw this as a way to avoid the expensive overage charges and let my other children have texting, too. This is when I began to experiment on my own.

Well, long story short, I now do a lot of texting! It is the primary communication I have with my two daughters who are away at college. Between their schedules and mine, it's difficult to find time to talk and they aren't constantly checking e-mail. We can send a text with a quick question or note and get a response when it is convenient. The students that I tutor prefer this method of communication to set up tutoring sessions. I have increased communication with several relatives and friends through texting. Even my 74-year-old father is now texting with his children and grandchildren, when he would normally only talk to some of them

Section Specialist

Sam Phillips, Staff Editor

during visits, holidays, etc. The more frequent texting has enriched the conversations that they do have when they see each other.

Why am I telling you this story? It is an example of why it is important to keep an open mind to new technologies and to not be afraid of experimenting with them. Another example is social networking, like FaceBook. My children are all on it, but I don't want to be. I don't have the time and don't see the need or benefit to me. I then realized that this sounded similar to my original texting argument. Therefore, I decided to sign-up for LinkedIn, a professional network. I made a bunch of connections, joined some groups and am exploring the benefits. It is clear to me that as these networks grow, new benefits to me and other actuaries will emerge. I encourage everybody to do this. The more of us out there exploring, experimenting and sharing, the faster the benefits will be realized. The SOA Technology Section has a group on LinkedIn, as well as the SOA and many other actuarial organizations. There are a lot of actuaries out there using LinkedIn, don't be left behind.

I'll be honest, I don't yet see huge benefits to me professionally, but I have talked to others who utilize LinkedIn and are reaping benefits. This might be because I am now getting older. Many younger actuaries are finding more benefits, but they have grown up during the FaceBook age and better understand how to utilize the network to exchange knowledge. As usage expands, there are more benefits to all, even us "older" actuaries.

I encourage everybody to set up a LinkedIn account, join the SOA Technology Section group and to start exploring. The more of us using it, the more benefits to all and we can grow with this technology. Keep an open mind to new technologies and we may all reap currently unknown benefits.

Please feel free to contact me or any Technology Section Council member with any ideas of how we as a section can serve your needs.

I just want to finish with a call for volunteers. Please contact me, Howard Callif or Eddie Smith if you are willing to submit an article for a future issue of CompAct. Also, we are looking for any volunteers who would like to serve on next year's section council. Elections will be held in the late summer. Let me know and I will be glad to answer any questions you might have.

Actuaries
Risk is Opportunity.®

475 North Martingale Road, Suite 600 Schaumburg, Illinois 60173
Phone: 847.706.3500 Fax: 847.706.3599 www.soa.org

ISSUE 35 | APRIL 2010

SOCIETY OF ACTUARIES Technology Section

CompAct

ELECTRONIC NEWSLETTER

CONTENTS

QUICK LINKS

Technology Section Web site

Council

Links of Interest

Fiction Contest

Howard Callif, Editor

SOA Staff
Meg Weber, Staff Partner

Sue Martz,

Share Print Article Search Back issues

EDITOR'S NOTES

by Howard Callif

People have submitted feedback on our new online format. Much of it is positive, but there is a group of people that want the newsletter in a printed format. We are looking into adding functionality to the Web site so all of the articles can be printed, so everyone can view the newsletter in the format that is convenient for them.

There are also other suggestions we've gotten to increase usability, and we are investigating these. Thanks to everyone who has provided feedback. I encourage you to submit comments and suggestions!

We have ideas for future columns, but need people to help out. We want to produce a "round up" of programming languages, comparing and contrasting them. We will try to demonstrate solutions to the same problem in SAS, APL, R, Excel, Visual Basic, C#, and any other language we can find volunteers to program in (F#, Perl ...). Please consider volunteering!

We are always looking for people to submit articles–please consider sharing your wisdom, learning experiences, or providing information on how you use technology to accomplish your goals.

Section Specialist

Sam Phillips, Staff Editor

Actuaries
Risk is Opportunity.®

475 North Martingale Road, Suite 600 Schaumburg, Illinois 60173
Phone: 847.706.3500 Fax: 847.706.3599 www.soa.org

ISSUE 35 | APRIL 2010

SOCIETY OF ACTUARIES Technology Section

CompAct

ELECTRONIC NEWSLETTER

CONTENTS

QUICK LINKS

Technology Section Web site

Council

Links of Interest

Fiction Contest

Howard Callif, Editor

SOA Staff
Meg Weber, Staff Partner

Sue Martz,

Share Print Article Search Back issues

HORSES FOR COURSES

by Phil Gold

Editor's note: This article originally appeared in the January 2007 issue of CompAct.

We live in an environment where the word open has positive connotations, while closed has a negative feeling to it. Linux is open source, and Windows is proprietary or closed. So Linux must be better, right? Well, for some people it is, and if you read the Internet blogs, there's no contest. Yet Windows has the bigger market share. How many of us are running on Linux today? Strangely, Apple's OS/X gets even better reviews and that is a closed system. So let's keep an open mind.

There is no universal answer that open code is better than closed code, or vice versa, although it seems almost an item of religion for some people. You have to look at the requirements of the application, the quality of the vendor, the size of the organization, initial and ongoing costs of each approach, corporate governance control requirements, the rate of change in the environment and the availability of skilled resources.

I am a software developer and I have chosen an approach closer to the closed end of the spectrum than the open end. I thought you might like to know why I made that decision, and what I have done in my application to meet the requirements for flexibility that many claim can only be satisfied by open code.

The first problem is to define my target market. Let's say my target

Section Specialist

Sam Phillips, Staff Editor

market is all life actuaries, everywhere. I am writing a system to perform all manner of actuarial calculations, first to support the requirements within my own country, then internationally. By far the easiest solution for me is to write an open code system. All I need to do is develop a nice framework, probably based on Excel or something that looks like it, add some database support and a report writer, then prototype some typical products and let the client or a consultant worry about adjusting the sample code to fit the real world products. Then I could look forward to a lucrative stream of consulting assignments to implement and maintain your systems. In fact, I must be nuts not to have followed this model. Why? Because the alternative is to code every possible combination of product features and regulatory requirements myself, as well as user-specific methodologies and approaches, and that would take forever. And yet, masochist that I am, that is indeed the path I chose. I'll tell you how later. Right now, I'll concentrate on why.

Why take the closed code route?

- Because I've been down the open code route before, in three companies. In every case each new product resulted in a new model with new source code, incompatible with other models in the company, and a variety of errors because of the lack of a proper testing environment. I want a reliable universal model, not a collection of independent models.
- Because I believe actuarial resources are scarce and expensive, and they should be employed on real actuarial problems, not developing software. Those actuaries that want to develop software should come and work with me–I'll need them for sure.
- Because despite the unique characteristics of each company and product, there is a lot more common ground between us than elements that separate us.
- Because I like a challenge. When someone tells me it can't be done, that's when I get interested.
- Because I was working in a reinsurance company, where we required a quick turnaround on each new product and the volume of such products prohibited the down time of developing a new model each time around.

Why persevere?

- Because the senior management of my company encouraged my efforts.

- Because my initial efforts were met with surprising success in the market.
- Because I found partners and employees who shared my philosophy.
- Because our clients gave us encouraging feedback.

Now the golden rule in software is do not bite off more than you can chew. The whole industry is tarred by those who promise the world and don't deliver. We developed a different philosophy. We never promised something unless we were certain to achieve it, and we encouraged potential customers to simply try what we had right now, and see if it would be useful to them. I would urge all in the industry to follow this model. This way, you get clients who trust you, and recommend you to their friends. We have lost new business along the way by refusing to make aggressive promises, but I am convinced it is the right way to proceed.

We proceeded by concentrating on particular market segments. We could not be all things to all people, especially at first. So we built out our portfolio gradually–first conventional products, then UL, then Par, then Disability, then Assets and so on, at the rate of about one new product line or module per year. We started off with just one country, then two, until today we operate on five continents.

Most of all, we concentrated on keeping our current customers happy. When you write closed code, you are taking on the responsibility of providing good service and fast response time. In our system there is only one code base and everyone gets the same software. So every client gets the benefit of each new feature no matter where the request came from.

OK, there's a problem right there–what about secret new features you don't want your competition to know about? This does happen, although as you know, there are few secrets in this industry. Suffice it to say there are various ways to solve this problem, and we can build in enough flexibility so that proprietary product features will not be given away by the software.

This is where the closed code approach pays off in a big way. If you have just one code base, then you can have many users reviewing and validating the calculations. It helps enormously when regulators and consultants review your software and during acceptance testing at each new client. If you have something wrong or missing in your code, you're going to find out and have a chance to fix it. This simply is not the case for open code. Think for a minute about the SOX implications here, their importance simply cannot be overestimated. Actually this is not a consequence of closed code but of common code. By going open code you simply preclude this level of scrutiny.

If you ask actuaries what is the biggest problem they face with their software, the most common answer will probably be the problem of keeping it up to date, the problem of conversions. Let's examine how these work on a completely closed system and a completely open system. On a completely open system, the vendor can really only provide changes to the framework and some new sample code. Changes to the framework must be limited or they will disrupt the current implementations. So the onus is on the developer to come up with the perfect framework for all time on day one. Let's be honest here. How many system architectures from 15 years ago are currently state of the art? If you got something wrong on day one, you can't always fix it later because users will have built their application around your architecture–change it and it breaks their applications. Now in the closed code context, we can provide automatic conversions of user models no matter what changes we make to the architecture. We have changed from DOS to Windows, from Basic to C++, from FoxPro to JET, and from a separate system for each country to a unified system, all without breaking the users' applications. Sure it takes a lot of work on our part, but the advantages are overwhelming. Users get a system that can be kept up-to-date with all the latest technology and functionality without massive conversion problems. They can upgrade in hours and not months. This is one of the biggest justifications for the closed code approach.

Now I promised to tell you how we tackle the need for flexibility, given we don't have infinite resources. Actually, it really does take an enormous effort to build in all the features clients require, and we have a very large and highly skilled workforce here we would not need if we were an open code shop. We have found that in most areas of the actuarial system we can provide enormous flexibility through the switches, scalars, tables and objects we have built up over the years, some at our own instigation and some to meet user request (about a 50:50 split). But there are some types of logic that are very hard to accommodate in this way. Take policyholder behavior for example, or crediting rate strategies or experience refunds.

Some closed code systems are really closed down tight. Others allow you insertion points, and with the aid of a compiler or by using a non-compiled language, they allow you to change the source code. The first option seems too rigid to me, although it is what we offered for a number of years because we felt the second option simply negates most of the advantages of closed code systems. Then one day our developers came up with a third way that provides the advantages of open code wherever you need it most, but preserves the integrity of the source code and allows for full automatic conversions between software releases. I won't go into too much detail, but the breakthrough involves treating user code as data input to the system, and some very sophisticated use of the .NET framework. The cost of having this type of expertise in house may be prohibitive for the end user.

If I were starting over today, would I do the same again? I'd have to say that from a financial and marketing point of view maybe not. It is cheaper to develop an open code system and probably easier to sell. But when I think of the client's best interest, I would have to say yes. For less ambitious software projects where the scope is more focused, the balance may be completely different. Each case should be taken on its own merits. So the golden rule is there is no golden rule. Choose instead the right horse for your particular course.

Phil Gold, FSA, MAAA, FIA, is a founding partner of GGY AXIS. He can be contacted at phil.gold@ggy.com.

ISSUE 35 | APRIL 2010

SOCIETY OF ACTUARIES Technology Section

CompAct

ELECTRONIC NEWSLETTER

QUICK LINKS

Technology Section Web site

Council

Links of Interest

Fiction Contest

Howard Callif, Editor

SOA Staff
Meg Weber, Staff Partner

Sue Martz,

Share Print Article Search Back issues

VISUAL BASIC NET REFERENCES

by Howard Callif

This article will provide useful references for learning and using Visual Basic NET. If you are interested in learning more, or want more articles on this subject, please let me know.

The VB Net 2005 development platform is much more difficult to pick up and start using than Quickbasic or VB6, so there may not be a large number of actuaries using the language. The 2008 development environment is significantly better, but unfortunately still has a much larger learning curve than BASIC from years past. Even so, I consider NET to be a premier business user tool. It is powerful and packed with functionality, and worth some extra setup time. The development environment is a platform for a large number of NET languages, so I think this will grow in popularity, and usefulness, since more languages will be added. Note that Visual Studio 2010 will be released in March, but I have not previewed release. Here are some useful reference books and sites, with summaries included. Since the new release is coming so soon, there may be updated versions of these books as well.

Programming Microsoft Visual Basic 2005: The Language, by Francesco Balena–Microsoft Press, 2006
This is THE book to buy on Visual Basic NET (in my opinion). It has extensive examples, covers a large spectrum of the language, and is an incredibly useful reference book. It is almost 1000 pages, but mine came with a CD that had an electronic copy. I wasn't able to verify whether the version on sale now still includes it.

Section Specialist

Sam Phillips, Staff Editor

Visual Basic 2005 in a Nutshell by Tim Patrick, Steven Roman, Ron Petrusha, and Paul Lomax–O'Reilly

There is an entire "In a Nutshell" series, and this is an excellent reference book series. There is a version for VB6 and VBA (Excel macros), and other languages. It includes a language summary, sample code, and identifies common errors and unexpected behaviors. This book will help you get up and running quickly for any language.

Practical Guidelines and Best Practices for Microsoft Visual Basic .NET and Visual C# Developers, by Francesco Balena and Giuseppe Dimauro–2005, Microsoft Press

This book helps explain not only what best practices are, but why. There are often times when a conflict arises, and implementing a "best practice" causes additional work, or other issues. There are also many practices that are simply a matter of preference, but one standard must be chosen for consistency across a team. This book identifies options and tradeoffs, and presents the pro's and con's for each.

Howard Callif, ASA, MAAA is product champion at Milliman IntelliScript. He can be contacted at howard@callif.org.

Actuaries
Risk is Opportunity.®

475 North Martingale Road, Suite 600 Schaumburg, Illinois 60173
Phone: 847.706.3500 Fax: 847.706.3599 www.soa.org

ISSUE 35 | APRIL 2010

SOCIETY OF ACTUARIES Technology Section

CompAct

ELECTRONIC NEWSLETTER

CONTENTS

Table of Contents

Letter From The Chair

Editorial

Horses For Courses

Visual Basic NET References

R Corner-Graphics

Number Puzzle

Articles Needed

QUICK LINKS

Technology Section Web site

Council

Links of Interest

Fiction Contest

Howard Callif, Editor

SOA Staff
Meg Weber, Staff Partner

Sue Martz,

Share Print Article Search Back issues

R CORNER–GRAPHICS

by Steve Craighead

Editor's note: R Corner[1] is a series by Steve Craighead introducing readers to the "R" language used for statistics and modeling of data. The first column was published in the October 2008 issue, and explains how to download and install the package, as well as providing a basic introduction to the language. Refer to each CompAct issue since then for additional articles in the series. The introductory article can be found on p. 24 of the October 2008 issue on the SOA Web site.

The graphical capabilities of R are extensive. R is truly an object-oriented language with classes built upon classes. Most classes that are available in R and its associated packages already have graphical methods. The majority of these are simply the standard method "plot." This method may be all that you need. For example Markus Gesmann has designed the ChainLadder package to determine insurance claims reserves. You can examine how R can be used graphically by installing the ChainLadder Package using the various Package options from R.

After loading the package, use the following commands (borrowed and modified from the Bootstrap-Chain-Ladder Model help screen):

```
B <- BootChainLadder(RAA, R=999, process.distr="gamma")
B
#Here is the ubiquitous plot() method
plot(B)
```

Section Specialist

Sam Phillips, Staff Editor

```
# Compare to MackChainLadder
MackChainLadder(RAA)
quantile(B, c(0.75,0.95,0.99, 0.995))
```

See Figure 1 for the result of these commands.

To demonstrate a few of the other capabilities of graphics, we will plot the empirical density of the total IBNR and the density of the data fit to a LogNormal distribution use these commands:

```
#Notice how the title is added by the "main" variable in the plot command
plot(density(B$IBNR.Totals),main="Empirical IBNR Density with LogNormal")

# fit a distribution to the IBNR library(MASS)

fit <- fitdistr(B$IBNR.Totals[B$IBNR.Totals>0], "lognormal")
#display the fit results to the R window
fit

#now let's layer-on the fitted lognormal curve onto the graph. Observe the use
#of "col=red" to plot the lognormal distribution and the "add=TRUE" to overlay
#the curve on the empirical density
curve(dlnorm(x,fit$estimate["meanlog"], fit$estimate["sdlog"]), col="red", add=TRUE)
```

R displays the fit statistics:

```
meanlog            sdlog
10.821176062       0.374878187
( 0.011860621)     ( 0.008386725)
```

Figure 2 displays the empirical distribution with the LogNormal distribution fitted to the data.

To change the axis labels, you use the "xlab" and "ylab" commands within the plot command as shown here:

```
plot(density(B$IBNR.Totals),main="Empirical IBNR Density",xlab="Sorted IBNR Data",ylab="Probability")
```

See Figure 3, for the results

If you want to add a legend to the graphic you can use these commands:

```
#replot the original plot(density(B$IBNR.Totals),main="Empirical
IBNR Density",xlab="Sorted IBNR Data",ylab="Probability")
#add the overlay
curve(dlnorm(x,fit$estimate["meanlog"], fit$estimate["sdlog"]),
col="red", add=TRUE)
#add the legend
#the first two numbers give the location of the legend within the plot
box, since the x values range from 0 to over 150K #and the y values
range from 0 to just over 2.0e-5, the legend will fit on an empty area
of the plot
#the following c("Empirical","LogNormal") list gives the elements in
the legend
#the two setting for col=c(1,2) will make sure that the line style will
have the colors corresponding to the plot.
#the lty=c(1,1) sets the line type of the graph to be a solid line. Other
line types can be used for dashed lines, etc.
legend(100000,0.000015, c("Empirical", "LogNormal"), col =
c(1,2),lty=c(1,1))
```

Now observe the legend in Figure 4.

For further examples and features that can be set with the legend command type:

```
library(help="legend")
```

For finer control you can control any component on the graphic, by changing other features documented for the "plot" command or by using the "par" command. Especially pay attention to the "mfrow" setting in the par command. For instance if you execute "par(mfrow=c(2,2))" you can place 4 graphs on one graphics device like in figure 1 above. "par(mfrow=c(1,1)" restores the graphic device to the original setting.

If you want to layer two separate "plot" commands on top of each other, use the "par(new=TRUE)" command between the two plots (this forces the second plot not to clear the graphic. To make both plots have the same xy ranges, you would use the "xlim=c(,)" and "ylim=c(,)" construct in both plots.

For my last example, I will limit the display of the data on the x-axis by using "xlim" command:

```
plot(density(B$IBNR.Totals),main="Empirical IBNR
Density",xlim=c(50000,150000))
curve(dlnorm(x,fit$estimate["meanlog"], fit$estimate["sdlog"]),
col="red", add=TRUE,xlim=c(50000,150000))
```

Figure 5 Restricted X Axis

Graphic Resources

As I said before, the graphical capabilities of R are extensive and I have barely scratched the surface. In this last section, I'll leave you with several excellent additional resources that lead you to discover more of the capabilities of R.

J. H. Maindonald has written an excellent article titled, "Using R for Data Analysis and Graphics Introduction, Code and Commentary." It is available here. In it he instructs the reader on the fundamentals of R as well as demonstrates the graphics capability of R.

Nicholas Lewin-Koh maintains the Graphic Displays & Dynamic Graphics & Graphic Devices & Visualization CRAN Task View. On this web site, he outlines the major topics of how R handles graphics, for both the user and the developer.

The home page of the R Graph Gallery is here. You can enter the gallery by choosing one of four different collections on the right-hand side of the Web page. Once you choose a gallery and then choose an example graphic, the Web page will display the graphic and under Requirements on the left-hand side of the screen, you can choose to either download the associated R code or examine the code in your Internet browser.

The last resource that I recommend is the R Graphics Manual, which demonstrates the collection of R graphics from the extensive list (now over 2000) of all R packages. The link is here.

Conclusions

This article is the last basic article on R. In future articles I will model various actuarial models or discuss the use of especially promising R packages. My next article will be a toy ERM model demonstrating the use of R in modeling copulas for the aggregation of stand-alone capital distributions to determine the diversified (economic) capital distribution at the enterprise level.

Steve Craighead, ASA, CERA, MAAA is a consultant at TowersWatson. He can be contacted at steve.craighead@towerswatson.com.

[1] Craighead, S. (2000), "Insolvency Testing: An Empirical Analysis of the Generalized Beta Type 2 Distribution, Quantile Regression, and a Resampled Extreme Value Technique," ARCH, pp. 13–149.

Actuaries
Risk is Opportunity.®

475 North Martingale Road, Suite 600 Schaumburg, Illinois 60173
Phone: 847.706.3500 Fax: 847.706.3599 www.soa.org

ISSUE 35 | APRIL 2010

SOCIETY OF ACTUARIES Technology Section

CompAct

ELECTRONIC NEWSLETTER

Share

Print Article

Search
Back issues

CONTENTS

Table of Contents

Letter From The Chair

Editorial

Horses For Courses

Visual Basic NET
References

R Corner-Graphics

Number Puzzle

Articles Needed

QUICK LINKS

Technology Section
Web site

Council

Links of Interest

Fiction Contest

Howard Callif, Editor

SOA Staff
Meg Weber, Staff Partner

Sue Martz,

ARTICLES NEEDED FOR COMPACT NEWSLETTER

Your help and participation are needed: Articles, book reviews and other information on technology useful to actuaries are needed! All articles will include a byline to give you full credit for your effort. An annual prize is awarded for the best article of the year. If you would like to submit or discuss an idea for an article, please contact Howard Callif (Howard@Callif.org).

Submission deadlines are May 15 for the July issue, and August 15 for the October issue.

Thank you for your ideas!

Section Specialist

Sam Phillips, Staff Editor

Actuaries
Risk is Opportunity.®

475 North Martingale Road, Suite 600 Schaumburg, Illinois 60173
Phone: 847.706.3500 Fax: 847.706.3599 www.soa.org