SOCIETY OF ACTUARIES

Article from:

CompAct

April 2008 – Issue 27

Standard File Format for Economic Scenarios to be Exposed for Comment

by Steve Strommen

The times they are a'changing. Years ago no one would have dreamed of offering all the guaranteed living benefits on variable life and annuity products that are available in today's marketplace. And who could have guessed that a mark-to-market model for public accounting might soon replace GAAP? These and other revolutionary changes in the insurance marketplace are placing greater emphasis on the use of stochastically generated economic scenarios for use in product pricing, financial reporting and risk management.

Financial models, implemented in software, are essential tools for the actuary in dealing with the new world, but as yet there are few standards regarding those tools. In terms of standards development, we are still in the days before the U.S. revolutionary war, when all rifle parts were made by hand.

Your Technology Section has recognized the potential benefits of standardization in this area and has assembled a team of volunteers working towards a standard approach to encoding economic scenario data. The hope is that collections of stochastically generated economic scenarios will someday be exchanged as easily as the MP3 files that are already the ubiquitous medium for digital music. Actuaries should not need to deal with the details of formatting scenario information for consumption by their computer models, and widespread adoption of a standard file format can free them from such drudgery.

After over a year's work, the project team has developed a prototype file specification, a definition for a software interface, and an implementation of that interface for exposure to the membership. This article discusses the goals that were set for that work and provides an overview of the results so far.

Specific goals for the standard scenario file format project are:

1) **Platform neutrality**—reflecting a need to work with any operating system, including Windows, Mac, Linux, or various mainframes or minicomputers.
2) **Comprehensiveness**—reflecting a need to contain most of the commonly occurring components of economic scenarios used in actuaries' practice areas, including insurance, investments, employee benefits, as well as maintaining a global perspective and allowing for currency exchange rates and economic conditions that vary by country.
3) **Extendibility**—as forecasting techniques incorporate additional types of models, the standard should not need to be completely reworked; further, software that uses the data in standard-adherent instances should be minimally affected by changes to the standard.
4) **Readability**—facilitating testing as well as examination by actuaries, data which maximizes understanding through visual inspection is desired.
5) **Simplicity of processing**—the standard will be more easily and happily accepted if software modules required to access data are readily available and have an intuitive interface.

(continued on page 24)

Steve Strommen is senior actuary with Northwestern Mutual. He can be contacted at stevestrommen@northwesternmutual.com

The choices considered for the file format include a new XML dialect, a proprietary binary format, and alternative textual formats. The team has chosen to use XML because it satisfies the list of goals better than the alternatives. XML has other properties useful and relevant to the representation of multiple economic scenarios. It is inherently hierarchical; it facilitates creation of self-documenting products, it allows vector-like structures for data that repeat by calendar month and year, and when necessary, it can be designed to be insensitive to ordering of data. As a text-based format, XML is also inherently readable.

The team has chosen to give the newly-invented dialect of XML a name: ESML, for Economic Scenario Markup Language.

The team has chosen to give the newly-invented dialect of XML a name: ESML, for Economic Scenario Markup Language. A full definition of ESML is beyond the scope of this article. A special addition to the Technology section Web site is being prepared to introduce ESML and provide complete documentation and a set of tools for working with ESML. The tools will include a library of routines callable from Excel VBA, and a sample application (scenario generator) built in Excel VBA. During 2008, all of this material will be in the nature of an exposure draft, and your comments and suggestions will be sought. By the annual SOA meeting a final version might be ready.

The remainder of this article describes three of the most interesting aspects of ESML. These are: 1) the kinds of information that can be included in the file, 2) dealing with potentially huge files, and 3) an approach for storage of yield curves.

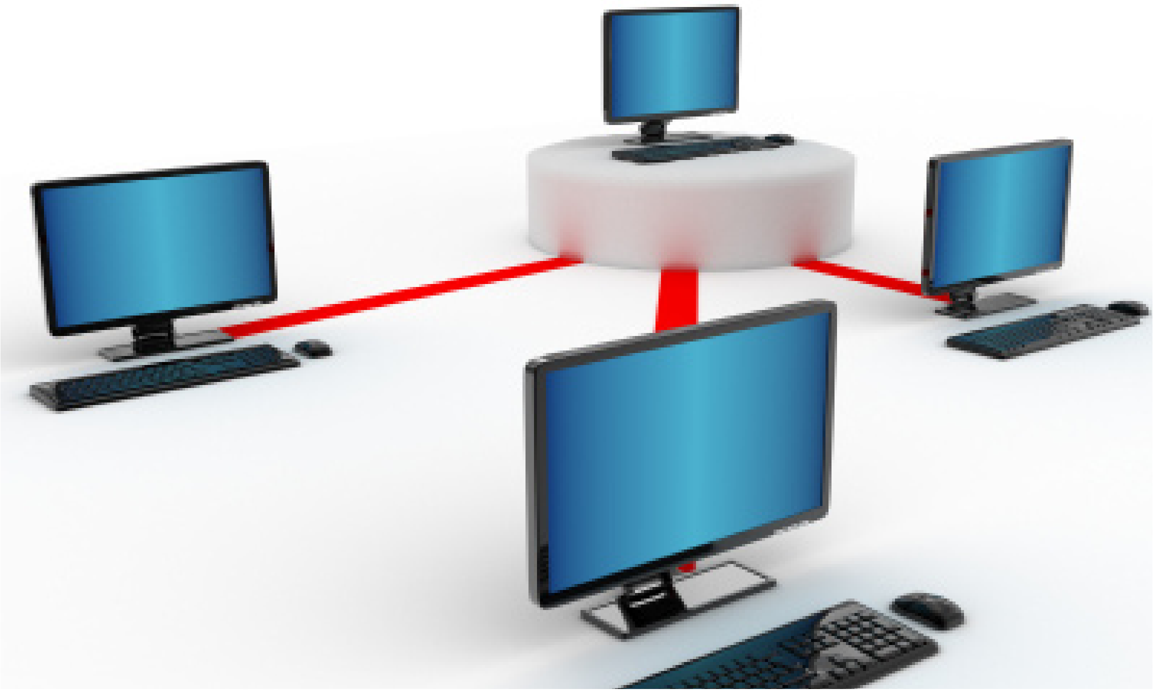
The kinds of information included in ESML

ESML documents contain two main sections: 1) information about the list of scenarios, and 2) the list of scenarios.

The first section of the file (containing information about the list of scenarios) contains items such as the name of the person creating the file, the generator used, any seed value for a random number generator, and a date and time of creation. Also, free-form text comments can be included to describe the intended purpose of the file or document anything special about its creation, such as parameter values used in the generator.

The second section of the file (containing the list of scenarios) contains as much as the generator puts in it. This can include, for any country on any date, any or all of the following:

- Several yield curves varying by credit quality.
- Investment returns for various kinds of equity investments or managed portfolios.
- Currency exchange rates.
- Other economic indexes or rates such as those for inflation, unemployment, GDP growth, and so on. One can include any index or rate by specifying a name for it and giving it a value.



A list of the kinds of information included for each country and date (as well as the list of countries) is included in the first section of the file as part of its descriptive information about the scenarios. Note that ESML uses ISO standards when appropriate for things like the exact spelling of codes for countries and currencies.

Dealing with large files

Collections of economic scenarios can be very large files. This can pose problems due to limitations on the amount of memory available. If the entire collection is not stored in memory, there is still a need to enable fast access to any individual scenario.

The ESML standard deals with this by associating a second index file with the XML file containing the scenario data. The index file contains the offset in bytes from the beginning of the scenario file where each scenario's data begins. The information in the index file can be used to allow software to keep as little as one scenario in memory at a time while still maintaining fast access to any scenario in the collection.

Efficient storage of yield curves

Yield curves have complex shapes, and a large amount of data can be required to fully describe a yield curve. Note that there are three fundamental representations of each yield curve: as forward rates, as spot rates, or as bond coupon rates. Any of those three curves can be converted to any other using software. However, one curve alone can involve 360 values if monthly maturities from one month to 30 years are used to define the curve.

Storing every point on the curve is just one way to store a yield curve. For some applica-

(continued on page 26)

Standard File Format for Economic … • continued from page 25

tions, that degree of accuracy is not needed, and fewer points might be stored, with some sort of interpolation being used between points. And for some applications, a formulaic or parametric description of the yield curve might be best.

ESML allows yield curves to be stored in any of these ways. This is accomplished by requiring each yield curve to be associated with a factory name and a set of stored values, each with its own name and value. The factory name is really the name of the algorithm used to reconstruct the full yield curve from the set of named values that are stored. For example, an algorithm that does simple linear interpolation between a few stored points might name each point with its maturity in months and might use a factory name of LinearInterp.

ESML allows any factory name and any names for the stored values representing a yield curve. However, only a handful of factory names will be recognized by any software implementation, and each factory will recognize only specific names for stored values. So as part of the ESML standard, the following factories are defined and should be implemented in any software that claims to implement the standard. (Additional factories might be added during the exposure period).

The BondData yield curve factory

In this factory the stored values are par coupon yields for semi-annual coupon bonds of 0- to 360-months to maturity. A total of 361 values are stored—a complete representation of the bond yield curve.

The string associated with each stored value is dxxx where the xxx represents the number of months to maturity (from 000 to 360).

The string ID for this kind of yield curve factory is BondData.

The Nelson-Siegel yield curve factory

In this factory the stored values are the parameters for a Nelson-Siegel curve that represents the spot rate curve by time to maturity. Each stored value represents a parameter of the curve. The parameters are:

b0 = the ultimate long term rate.
b1 = the excess of the instantaneous short rate over the long term rate (normally negative).
b2 = a parameter governing the size of hump in the yield curve.
k = a parameter governing the location of hump in the yield curve.

The spot rate for t years to maturity is calculated as follows. Of course t can be fractional when calculating yields for a number of months that is not a whole number of years.

spotRate(t) = b0 + (b1+b2/k)*((1-Exp(-k*t))/(k*t)) – (b2/k)*Exp(-k*t)

The string associated with each stored value is the name of the parameter as given above.

The string ID for this kind of yield curve factory is Nelson-Siegel.

The LinearInterp yield curve factory

In this factory the stored values are a small number of points on the bond curve. Each point represents a different number of months to maturity. The remaining points on the bond curve are interpolated or extrapolated linearly. This can lead to a somewhat saw-toothed pattern of spot rates and forward rates, but often that is not viewed as a material problem.

The string associated with each stored value is dxxx where the xxx represents the number of months to maturity (a few values from 000 to 360).

The string ID for this kind of yield curve factory is LinearInterp.

The NSInterp yield curve factory

In this class the stored values are two or three points on the spot curve. Each point represents the spot rate for a different number of months to maturity. The remaining points on the spot curve are interpolated by fitting a Nelson-Siegel curve to the points that are provided.

If the number of stored values is two, then the yield curve has a typical curved shape. The values of parameters b0 and b1 are solved for with the assumption that b2 = 0 and k = 0.5.

If the number of stored values is three, then the values of parameters b1, b2, and b3 are solved for with the assumption that k = 0.5.

The string associated with each stored value is dxxx where the xxx represents the number of months to maturity (a few values from 000 to 360).

The string ID for this kind of yield curve is NSInterp.

Web site exposure on the way

As was mentioned earlier, a new page is being added to the Technology Section Web site to expose ESML work to the membership. Materials to be exposed will include:

1. A document outlining the file format.
2. A sample ESML file.
3. A software implementation of routines to read and write data in ESML, including all of the following:
   a. A Users Guide document.
   b. An online help file that serves as a detailed reference to the API.
   c. A compiled library usable in Windows, NET.
   d. Source code for the library in C#.
   e. A sample Excel VBA application that uses the library to read and write collections of scenarios.

An installation program will be available to facilitate installation of all this material on your PC under Windows. Two additional implementations of the software interface using standard C++ and Java are planned, but will not be initially available.

As was mentioned earlier, a new page is being added to the Technology Section Web site to expose ESML work to the membership.

If you work with stochastic economic scenarios, please take some time to review the ESML proposal and send comments. The Web page will indicate the e-mail address to which comments should be sent.