



MB Risk ManagementTM
Software - Consultancy - Training
an FSSTM - Financial Systems SoftwareTM company

Warnford Court
Throgmorton Street
London EC2N 2AT
United Kingdom
Tel: +44 20-7628 2007
Fax: +44 20-7628 2008
E-mail: mbrm@mbrm.com
Website: www.mbrm.com

RISK MANAGEMENT SYSTEM DEVELOPMENT USING LOW COST “ADD-IN TOOLKITS”

by Dr. Mamdouh Barakat

A perennial problem in software development is how to access common algorithms (e.g. analytical models for derivative pricing and risk management sensitivities) when working on different systems without re-inventing the wheel every time. Different systems may include (some of these systems may overlap):

- Spreadsheet modelling environment, e.g. using Microsoft Excel.
- Front Office Trading system
- Portfolio Management system
- Risk Management system, e.g. where the data is held in a database, e.g. Oracle, Sybase, Microsoft Back Office etc.
- Back Office system

The benefits from establishing common libraries of algorithms are numerous, including :

- Increased speed of development
- Reduced cost of development and maintenance
- Reduced chance of errors (since only one copy of the code is maintained)
- Lowers the learning curve when developers move from one system to another.
- The option to outsource the development of some of these algorithms (e.g. exotic option models) to specialist software houses. This could reduce costs, provide extra flexibility and implement new market tested analytical models.

Prior to the establishment of Microsoft Windows and enterprise computing, it was very difficult to set up such common algorithms because the base platforms were very different. For example the modelling environment may have been Lotus 1-2-3, while some front office systems may have been written in DBASE language, while the risk management system may have been running on a UNIX based system. In these environments, the same compiled code could not run, therefore internal standardisation based on a common source code library (e.g. “C”) was often used. However systems may be written in different programming languages (e.g. Cobol, Fortran, C, Pascal etc.). Also, the same source code compiled in different environments may behave differently (generally due to obscure bugs or implementation variations in the different compilers).

Low cost libraries from third-party software houses were also uncommon since, in the financial sector, there tends to be great secrecy on how algorithms are implemented (even if the underlying theory was widely published). Also, the different target environments for these libraries fragmented the market, reducing potential economies of scale.

The establishment of Microsoft Windows as a common environment, either stand alone or part of an enterprise wide client/server architecture, provided a key opportunity for third party software houses, such as MBRM, to develop a compiled libraries (“Add-in Toolkits”) of algorithms for the pricing, risk management, trading, fund management and auditing of securities, options, futures and swaps in the fixed income, commodities, equities, foreign exchange and money markets. The key technology to unlocking this potential in the Windows environment is Dynamic Link Libraries or DLLs for short. These are compiled libraries which can be called from numerous windows program, e.g. Excel, Visual Basic standalone, Visual Basic in Applications (VBA), C, C++ etc. This overcomes the problems in the older environments by providing :

- A common compiled library of routines which can be called from environments written in another language.
- Confidential source code is protected



MB Risk ManagementTM
Software - Consultancy - Training
an FSSTM - Financial Systems SoftwareTM company

Warnford Court
Throgmorton Street
London EC2N 2AT
United Kingdom
Tel: +44 20-7628 2007
Fax: +44 20-7628 2008
E-mail: mbrm@mbrm.com
Website: www.mbrm.com

- Linking of libraries with local or enterprise databases via the use of ODBC standard (Microsoft's Open Database Connectivity).
- Mass market enables economies of scale to develop.

To show how these "Add-in Toolkits" are called from different systems, an example is useful. In MBRM's "Universal Options Add-in", the "implied volatility" function has the following parameters:

1. Short term Interest Rate
2. Underlying Instrument Price
3. Strike Price
4. Option price
5. Years to expiry
6. Dividend yield (set to 0 if underlying instrument is a future)
7. Option Class (0 for puts, and 1 for calls)
8. Option Model (0 = Black Scholes, 1 = Binomial on Cash, 2 = Black, 3 = Binomial on Futures etc.).

In Excel, to calculate the implied volatility of an American style "at the money" 1 year equity index put option priced at 291, whose strike is 4500, with 6.5% short term interest rates, and a dividend rate of 3% :

=UOA_IMPLIEDVOL(6.5%, 4500, 4500, 291, 1, 3%, 0, 1)
Return 20%

As a convention, MBRM uses the first 3 letters of the function name to identify the add-in toolkit used (e.g. functions beginning with "UEA" would refer to the "Universal Exotics Add-in").

The "implied volatility" function in definition in C would be :

double UOA_IMPLIEDVOL(double dR, double dS, double dK, double dPRICE,
double dTime, double dDivRate, short Class, short Model);

The "implied volatility" function definition in Visual Basic would be :

Declare Function UOA_IMPLIEDVOL Lib "UOA_VB.DLL" (ByVal INT_RATE As Double,
ByVal INST_PX As Double, ByVal STRIKE As Double, ByVal OPTION_PX As Double,
ByVal YEARS_TO_EXPIRY As Double, ByVal EFD_DISCOUNT_RATE As Double,
ByVal CLASS As Integer, ByVal OPTION_MODEL As Integer) As Double

Excel, C and Visual Basic would all call the same compiled library code.

MBRM also bundle their libraries with some front end spreadsheets developed in Excel which call the common library (e.g. to perform portfolio revaluation and risk analysis). These spreadsheets, which can be linked to real-time feeds, are for users looking for "off-the-shelf" pricing and risk management software.

If a financial institution also requires source code libraries, then MBRM can provide, at additional cost, the source code for their Add-in Toolkits. This has the advantage that the financial institution would be buying in the tried and tested source code which is used by over 30,000 users world-wide.

This "economies of scale" of one library for all environments has enabled MBRM to substantially reduce its prices over the last few years and add increased functionality as standard.

In conclusion, I believe that "Add-in Toolkits" provide an open ended, flexible and low cost approach to assisting in building risk management systems. The ability to pick from, compare and combine libraries from different software houses is very beneficial.

XX

Dr. Mamdouh Barakat is Founder and Managing Director of MB Risk Management.