Proceedings of the 2004 Winter Simulation Conference
R .G. Ingalls, M. D. Rossetti, J. S. Smith, and B. A. Peters, eds.

VISUAL MODELING OF BUSINESS PROBLEMS: WORKFLOW AND PATTERNS

Lev Virine
Jason McVean

Calgary Technology Center
Schlumberger Ltd
600, 322-11 Avenue S.W.
Calgary, AB, T2R OC5, CANADA

ABSTRACT

Computer-based business analysis relies on models, or algorithmic representations of the business process. Real-life business problems can become very complex, which creates difficulties in generation, analysis, testing, and the actual use of the models. The paper discusses a proposed solution: the visual modeling workflow. A diagram or a group of diagrams represent each step within this workflow. The visual modeling process can be simplified by applying patterns or problem-solution formulas. Such modeling patterns include decoupling, encapsulation, visualization of user workflow, multi-layer visual representation of the calculation logic, and early identification and visualization of uncertainties. The patterns are applied to the visual modeling workflow, which include high level conceptual modeling, using Domain Models and Calculation Diagrams to visualize the calculation logic, visualization of testing and consolidations, and visualization of results of probabilistic analysis and simulation. The described methodology is used in a number of Schlumberger's software application.

1 INTRODUCTION

Effective analysis of complex business processes requires powerful and easy-to-use computer-based tools. This paper discusses only a limited group of business processes such as economic evaluation and analysis including fiscal modeling, decision and risk analysis, forecasting, scheduling, portfolio management, quantified market analysis, and others (Van Vliet and Hendry 2004).

There are at least three distinct approaches to the implementation of tools for business analysis. The first approach is based on spreadsheet applications. One of main advantages of spreadsheet approach is flexibility: Calculations can be implemented without creating a computer application; models can be easy updated and executed with different data sets; modeling in spreadsheets does not required advanced knowledge of computer programming; data can be easy visualized in standard tabular and graphical formats. Spreadsheet applications can also support scripting languages, which can be used for more complex models. Finally there is a number of business analysis add-ins built to operate seamlessly with spreadsheet applications.

Spreadsheets sometimes prove to be effective modeling tools for small and medium size business processes, but there are a number of problems associated with the use of them for complex business analysis. For instance, data and calculation algorithms are not presented simultaneously, making it difficult to understand and troubleshoot large spreadsheets. Furthermore, the visualization of relationships between variables in spreadsheets is limited Finally, spreadsheets applications have limited interface tools to design and present user forms for data input and output.

Another approach in implementing business analysis tools is to create specialized computer application for a particular model or a group of models. In this case the model or the interface associated with the model cannot be easily updated without updating the application. However, specialized applications are in most cases easier to use than spreadsheets because the user interface and calculation are optimized for the particular problem.

Finally the third approach to the implementation of tools for business analysis is the integrated solution framework. Such frameworks offer a flexible environment where business models can be easy created and maintained, user interaction is simplified, and data from multiple sources can be accessed seamlessly. There are number of large and comprehensive enterprise resource planning systems, such as SAP, PeopleSoft, Siebel, i2, and Oracle (Andersen 2003). These systems are capable of solving a wide variety of business problems. Such systems have a large number of components, which are built around a common framework based on common architectural concept. A solution can be built using these components as building blocks for the particular problem using customer-defined processes and data. Such solution frameworks are designed for complex repeatable processes within the organization. However, they are sometimes not efficient tools for stand-alone business analysis problems, especially if the framework is not yet enabled across the organization.

There is also another class of business analysis and simulation frameworks, which be can easy customized to solve different business problems. Examples of such solutions include Analytica by Lumina Data Systems, Extend by ImagineThat Inc. (Krahl 1999), GoldSim and GoldSim technology group, Enterprise Optimizer by River Logic Inc., and others. These tools allow the user to create a model of the business process, create a user form to input data and present the output, access data in different formats, and provide different kinds of simulations. These systems represent a significant step forward beyond spreadsheets and spreadsheet's add-ins. However, there are a number of problems associated with such systems (Paul and Serrano 2003). Each of the applications uses its own visualization approach to represent the model and simulation algorithm (Virine and Rapley 2003). Due to the lack of visualization standards, the modelers and the users can have difficulty reading and understanding the model created by another application. Although the applications mentioned above are focused on different problems, the modeling workflow is similar. However these applications represent and communicate the workflow differently, often using non-standard terminology. Finally the visualization of certain workflow steps such as calculation logic, testing, and consolidation remains a challenge.

To overcome the problem associated with business analysis and simulation frameworks, the scalable visual modeling workflow for business analysis problems is proposed here. Each step within the workflow is associated with a particular visual tool or visualization method. A set of rules can be applied to move from one step of the process to another. Visual modeling process can be simplified by applying patterns of tried-and-true solutions to modeling problems. Using the proposed workflow and patterns, the design and implementation of the models as well as the use for the models for analysis and simulation can be significantly simplified.

2 VISUAL MODELING PATTERNS

Patterns are named problem-solution formulas that codify exemplary modeling principles (Larman 2002). Patterns also can be referred to as tried-and-true solutions to modeling problems expressed as best-practice principles. Patterns are widely used in the software development and workflow management (van der Aaltst and van Hee 2004) as a way to capture best practices for design and analysis. However, patterns so far are not extensively used in modeling of business problems.

Some important patterns are common for different workflows and applications. Patterns are usually presented in the form of problem – solution description.

2.1 Decoupling

The problem is to simplify complex models. Such models are often comprised of multiple independent calculations, which are linked together and as such, are difficult to analyze and test. The solution is to analyze dependencies between calculations. It is important to know what variables are common between different calculations and what output variables are the inputs for the next calculation. The number of such links between calculations should be minimized. Each calculation should be tested separately if it is possible. Decoupling is closely related to visualization: each calculation can be presented separately on a visual diagram. For example, operating cost calculation includes many variables but only total operating cost time series would generally be used for the next step in the calculation of project's net present value.

2.2 Encapsulation

If the model has a large number of variables it complicates the analysis and testing. The solution is not to expose information about intermediate variables to other calculations. An example of encapsulation is the calculation of product revenue. Internal calculation of the revenue includes a number of intermediate variables. These internal variables will not visible or accessible from any other calculations.

2.3 User Workflow Visualization

Another problem in the modeling process is how to simplify the user interaction with the model, especially when model has a large number of variables with relationships to each other that are not obvious. The solution is to present the user with a system of steps which needs to be followed to use the model. This is a user or run-time workflow view. For example, the information about sequential calculation of sales, revenue, costs, and NPV can be presented within user interface in the form of a user workflow chart in the order that makes sense to the user.

2.4 Multi-Layer Representation of Calculation Logic

It is desirable to use visual tools to represent the calculation logic of a model. However, a single universal solution for simple visualization of calculation logic is not available. According to this pattern, calculation logic for one model can be represented by different views and each view can have multiple layers. Layers represent information which can be shown or hidden for the particular view. For example, a calculation algorithm with time series data can be represented by block diagrams in one layer and in a spreadsheet-type view with testing data associated with the calculation in another layer.

2.5 Early Identification and Visualization of Uncertainties

Most of business processes include analysis of uncertainties. Identification of uncertainties and their properties is an important but complicated process. The solution is to identify and visualize the uncertainties on all stages of the modeling workflow starting from on conceptual design of the model. For example if it was identified, that a capital cost is an uncertain parameter affecting net present value of a project, then the uncertainty in that capital cost should be visualized on calculation, testing, consolidation, and other diagrams.

3 OVERVIEW OF VISUAL MODELING WORKFLOW

The proposed visual modeling workflow, the design of which is based on the above patterns, is presented in Figure 1. The workflow to create a model includes three main steps. Extra steps are required for analysis and simulation of the model and depends on the particular problem. Each step is represented by the series of views. Each view can include one or more diagrams.

The first step is the high level modeling of the business process. The step includes: identifying problem and opportunities, asserting the business situation, generation creative alternatives, and providing a high level modeling of the situation (Howard 1988, Skinner 1999). The step can be represented by at least 3 views.

The conceptual view represents relationships between different calculations (processes) and simulation methods. It can be generic (high level relationship between processes), hierarchical (showing sub-processes and sub -models), time-based (for time related processes), and/or location-based (for location related problems). The data source definition view (DSDV) includes graphical representations of data sources, high level definitions of input and output variables and their relationship with calculations and simulation methods. Data sources can include databases, spreadsheets, application data inputs, and the results of previous calculations.

There are two types of workflow views: the modeling process workflow view, and calculation and simulation workflow view. The modeling workflow view is the mapping of the general workflow to the particular business process. This is the workflow for the modeler. The calculation and simulation workflow view is intended to be for the user of the model (see user workflow visualization pattern).

The second step in creating a model is the detailed modeling of the business process. This step can be represented by at least three views: a calculation script, a calculation diagram, and a domain model.

Some advanced business analysis and simulation frameworks use scripting languages to define calculation algorithms using easy-to-use notation. Scripting languages provide access to model variables and functions.

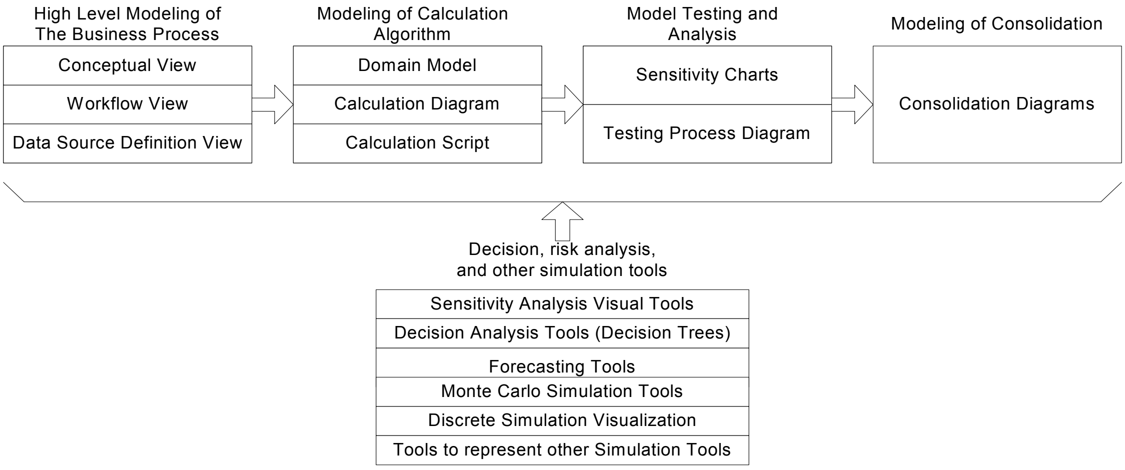
A domain model provides a structural decomposition of the model. It separates the model into its calculations and presents input and output variables for each calculation.

A calculation diagram is one of the most important concepts of visual modeling workflow. It is intended to visualize a calculation algorithm and its associated data. The domain model and calculation diagram will be discussed in more detail in Section 4. The third step of the workflow is the visualization of testing and analysis of the model. This steps includes two views: the testing process view (related to DSDV for the particular testing process), and sensitivity chart to ranges of input testing data together with ranges of outputs.

4 VISUALIZATION OF CALCULATION ALGORITHM

4.1 Domain Model

The most time-consuming step in the modeling process is the visualization of calculation algorithm. The domain



High Level Modeling of The Business Process
Conceptual View
Workflow View
Data Source Definition View
Modeling of Calculation Algorithm
Domain Model
Calculation Diagram
Calculation Script
Model Testing and Analysis
Sensitivity Charts
Testing Process Diagram
Modeling of Consolidation
Consolidation Diagrams
Decision, risk analysis, and other simulation tools
Sensitivity Analysis Visual Tools
Decision Analysis Tools (Decision Trees)
Forecasting Tools
Monte Carlo Simulation Tools
Discrete Simulation Visualization
Tools to represent other Simulation Tools

Figure 1: Visual Modeling Workflow

model is one of the views that is useful in this step. It is represented by a diagram with a number of boxes. Making use of the encapsulation and decoupling patterns, each box represents a calculation, global parameters, or inputs used in calculations. Each calculation box has two or three columns. The left column show calculation inputs, the right column show calculation outputs, and the middle column is optional and can show intermediate variables. Particular calculation formulas are not represented within a domain model. An example of a domain model for the economic evaluation of oil reserves is presented on Figure 2.

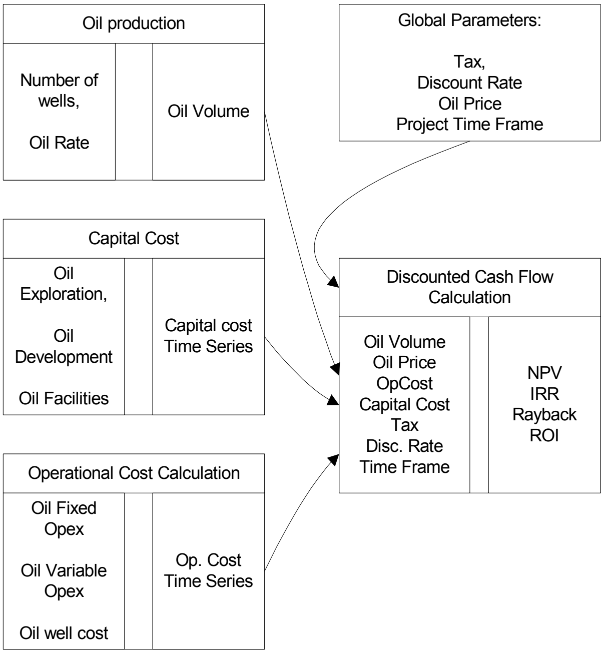


Figure 2: Domain Model Diagram

4.2 Calculation Diagram

The calculation diagram represents the calculation algorithm by combining the advantages of flow charts, spreadsheet representations, and scripting languages. It presents both the calculation algorithm and associated testing data together. The data within and calculation diagram is the testing data and used by the modeler to determine if the calculation algorithm is defined correctly. Calculation diagram also includes relationship between variables in form similar for block diagram. Because this is primarily an intermediate diagnostic tool, the end user may not necessarily see this view. Instead the user form or a dashboard can be used.

The calculation diagram can become very crowded, so the information can be separated into three layers (according to the "Multi-Layer Representation of Calculation Logic" pattern):

1. Calculation algorithm layer: Variables and calculation expressions
2. Variables relationship layers: Precedents and dependent arrows between input, output, and intermediate variables.
3. Data layer: Data associated with variables

Each layer can be presented separately or together with other layers.

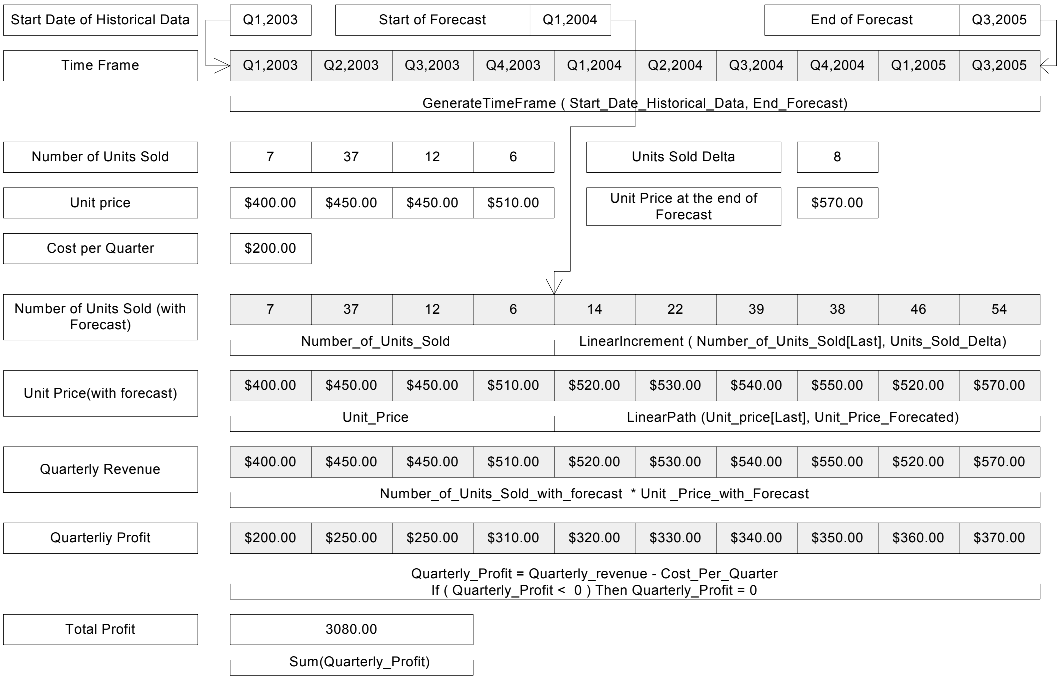
The calculation diagram is designed to operate with different data formats, or data shapes. Data shapes, as distinct from data types (discussed below) can include scalars, vectors, and matrices. Business analysis applications operate mostly with scalars and time series (vector) data shapes.

In business modeling definition of data types is much broader then in computer programming, where data type can be double, integer, etc. Data type in business models represents the nature of the variable. For example data type can be costs, revenues, prices, taxes, dates, etc. It is required as an additional level of control for the model. For example, if the modeler makes a mistake and adds units sold to a cost, the modeling framework will easy identify the problem as long as the units sold variable and the cost variable have different types. Data type analysis can be very important when consolidation of results of business analysis from different sources is required. Available data types can be predefined for the model or the group of the models.

Another characteristic of variables is that they can be input, output, or intermediate. Input variables do not have precedents. Output variables do not have dependants. Intermediate variables have both precedents and dependants, and are not presented to the user within a user interface form.

The process of the generation of a calculation diagram includes the following steps:

1. Define the global calculation parameters. This could include units, currencies, periodicity, tax rates, discount rates, etc. In the example of the calculation diagram shown in Figure 3, the calculation parameters are "Start Date of Historical Data", "End of forecast", and the quarterly periodicity. These parameters are used to define the time frame of the calculation, which is used for all subsequent calculations.
2. Define the input and output parameters. The purpose of the calculation shown on Figure 3 is to calculate total profit based on number of units sold, unit price, and cost per quarter. The calcula tion uses one-year of historical data plus forecasted data. Properties (such as external name of variable, internal name of variable, shape, and data type) for each variable must be assigned. Static elements of the diagram, such as rectangles with the external name of the variables can be defined. Input, output, and intermediate variables can be represented by different colors.
3. Define the calculation algorithm. The calculation is performed using standard and/or custom func-



Start Date of Historical Data	Q1,2003	Start of Forecast			Q1,2004			End of Forecast		Q3,2005
Time Frame	Q1,2003	Q2,2003	Q3,2003	Q4,2003	Q1,2004	Q2,2004	Q3,2004	Q4,2004	Q1,2005	Q3,2005
	GenerateTimeFrame (Start_Date_Historical_Data, End_Forecast)									
Number of Units Sold	7	37	12	6	Units Sold Delta			8		
Unit price	$400.00	$450.00	$450.00	$510.00	Unit Price at the end of Forecast			$570.00		
Cost per Quarter	$200.00									
Number of Units Sold (with Forecast)	7	37	12	6	14	22	39	38	46	54
	Number_of_Units_Sold				LinearIncrement (Number_of_Units_Sold[Last], Units_Sold_Delta)					
Unit Price(with forecast)	$400.00	$450.00	$450.00	$510.00	$520.00	$530.00	$540.00	$550.00	$520.00	$570.00
	Unit_Price				LinearPath (Unit_price[Last], Unit_Price_Forecated)					
Quarterly Revenue	$400.00	$450.00	$450.00	$510.00	$520.00	$530.00	$540.00	$550.00	$520.00	$570.00
	Number_of_Units_Sold_with_forecast * Unit _Price_with_Forecast									
Quarterliy Profit	$200.00	$250.00	$250.00	$310.00	$320.00	$330.00	$340.00	$350.00	$360.00	$370.00
	Quarterly_Profit = Quarterly_revenue - Cost_Per_Quarter If (Quarterly_Profit < 0) Then Quarterly_Profit = 0									
Total Profit	3080.00									
	Sum(Quarterly_Profit)									

Figure 3: Calculation Diagram

tions. Standard functions are commonly used calculations and manipulation. Custom functions are developed by the user using a scripting language. The relationship between variables is represented using connector lines with arrows. Connectors in current examples are used to represent the generation of the time frame based on three input dates.

4. Test the calculation algorithm. Because testing data and the calculation algorithm are presented within one calculation diagram, the analysis and testing of the algorithm is simpler. Each change in the calculation algorithm can be automatically reflected in the test data output. By graphing intermediate and output data from the calculation diagram the modeler can more easily identify the potential problems.
5. Generate user interface form (the dashboard). One of the main advantages of the calculation diagram is that user input form can be generated semi-automatically. The modeler generates an input form using calculation diagram by hiding the calculation algorithm, intermediate variables, and possibly connectors which represent relationships between variables. Also some input controls can be moved or updated to simplify user input. Static text and charts can then be added to the control.

To summarize, here are some advantages of the calculation diagram:

1. The calculation diagram presents data, calculations, and relationship between variables on the different layers within the same document. It simplifies the analysis of the model.
2. The calculation diagram is not a completely new way to represent the model. It is based on combination of standard flow chart format and spreadsheet-style data representations. These modeling techniques are familiar for most modelers. It significantly reduces a learning curve.
3. The calculation diagram significantly simplifies the operation with time series data, which are very common in business analysis.
4. The user input forms can be easily generated using the calculation diagram.

5 BUSINESS MODEL TESTING

One of main challenges in the modeling of the business problems is to provide the modeler testing and data analysis tools. Even with very advanced visual modeling tools, the modeler may have problems with the capturing of user requirements or make an error in the modeling process. Most business models can be very complicated which significantly increases the chance for errors. Such problems sometimes cannot be detected using limited sets of testing data. Analysis of multiple different combinations of input parameters is required.

Testing of the model can be presented using the testing process diagram which includes data sources and data ranges for all variables.

To test the model, the business analysis solution framework should provide calculations for all requested combinations of inputs ranges. An example of the results of a simple testing process is shown on Figure 4. The project NPV is calculated based on different data associated with variable "All Capital". All other variables will have base value. Known solution is presented as a red dot on the chart. The spike in the chart may be associated with an error in the model. Because possible number of requested calculations for all variables can be huge, the chart does not need to be always generated. But solution framework can inform the modeler about any unexpected non-linearity in the dependencies between input and output variables.

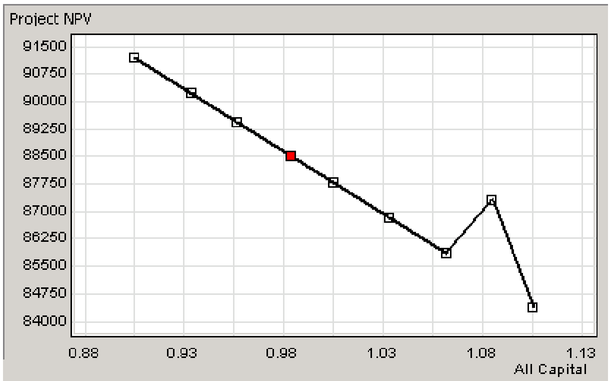


Figure 4: Results of Testing Process

CONCLUSIONS

This paper describes a proposed visual modeling workflow, which is a process of defining a model and calculation/simulation process through a sequence of steps, each of which is visualized by views. These views include a set of diagrams that represent model data, calculation algorithms and parameters, and the relationships between different calculations. Each step in the modeling process informs the next step in a way that increases the ease or creation and reliability of the resulting models. The workflow is based on modeling patterns or best-practice principles.

This methodology is being used in the development of new products in Schlumberger Information Solutions.

REFERENCES

Andersen, G., 2003. SAP Planning: Best Practices in Implementation. SAMS Publishing.

Howard, R. 1988. Decision Analysis: Practice and Promise. *Management Science*, 34: 679-695.

Krahl, D. 1999. Modeling with Extend. In *Proceedings of 1999 Winter Simulation Conference*, ed. P. A. Farrington, H. B. Nembhard, D. T. Sturrock, and G. W. Evans, 188-195. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

Larman, C., 2002. *Applying UML and Patterns*. 2nd Ed. Upper Saddle River, New Jersey: Prentice-Hall, Inc.

Paul, R.J. and Serrano, A., 2003, Simulation for Business Processes and Information Systems Design In *Proceedings of 2003 Winter Simulation Conference*, ed. S. Chick, P. J. Sanchez, D. Ferrin, and D. J. Morrice, 1787-1796. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

Skinner, D., 1999. *Introduction to Decision Analysis*. 2nd Ed. Gainesville, Florida: Probabilistic Publishing.

Van der Aalst, W. and Van Hee, K., 2004, Workflow management: Models, Methods, and Systems. Cambridge, Massachusetts: MIT Press.

Van Vliet, B. and Hendry, R., 2004, Modeling Financial Markets : Using Visual Basic.NET and Databases to Create Pricing, Trading, and Risk Management Models, New York, New York: McGraw Hill.

Virine, L. and Rapley, L., 2003. Visualization of Probabilistic Business Models. In *Proceedings of 2003 Winter Simulation Conference*, ed. S. Chick, P. J. Sanchez, D. Ferrin, and D. J. Morrice, 1779-1786. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

AUTHOR BIOGRAPHIES

LEV VIRINE is a senior software engineer of Schlumberger Information Solutions, a division of Schlumberger Ltd. He received his post-graduate degree from Moscow State University in 1986. His current research interests include decision analysis, risk management, and their applications for oil and gas industry. His email address is `<lvirine@slb.com>`.

JASON MCVEAN is the risk technology coordinator for the Schlumberger Information Solutions Value & Risk portfolio. He received his post graduate degree from the University of Calgary in 1994. His email address is `<jmcvean@slb.com>`.