Journal of Internet Banking and Commerce

An open access Internet journal (http://www.arraydev.com/commerce/jibc/)

Journal of Internet Banking and Commerce, August 2008, vol. 13, no.2
(*http://www.arraydev.com/commerce/jibc/*)

An Innovative Spreadsheet Authoring Environment

Richard J. Petti, PhD
President, ModelSheet Software LLC, Arlington, MA USA
Postal Address: **Richard Petti, ModelSheet Software LLC, 146 Gray Street, Arlington, MA 02476, USA**
Email: rjpetti@modelsheetsoft.com
Dr. Richard Petti is President of ModelSheet Software LLC, Arlington, MA. His areas of interest are algorithms, applications and software for business analytics. Most reports of his work in business analytics are proprietary information of McKinsey & Company, Booz Allen & Hamilton (and clients), General Electric Company, and The MathWorks, Inc. He has also published over a dozen papers in refereed journals in various fields of pure and applied physics.

Howard I. Cannon
CTO and Chairman, ModelSheet Software LLC
Postal Address: **Howard Cannon, ModelSheet Software LLC, 881 East First Street #501, Boston, MA 02127, USA**
Email: hicannon@modelsheetsoft.com
Mr. Cannon is CTO and Chairman of ModelSheet Software LLC. His current research interests are software architecture, user interface and implementation of web systems for business analytics. While working on the MIT Research Staff, Mr. Cannon invented Flavors, the first non-hierarchical object-oriented programming system. He has been president or vice president of several software companies.

Abstract

Spreadsheets are problematic components in enterprise information systems, due to reliability and other problems. The root problem is that the greatest strength of spreadsheets – enabling authors to work with visual layouts of models instead of

program code – is a double-edged sword, especially in more complex applications. Spreadsheets are simple in the sense that the set of primitive operations is small; yet they are not simple in the sense of having a sufficiently rich set of primitives for building complex models. What is needed in spreadsheet land is a new synthesis of visual layout and program structures that preserves the virtues of spreadsheets while introducing the missing structures – time series, dimensions, variables, data types – in a way that does not render business analysts dependent on programmers, as BI and ERP systems often do. This paper compares these design criteria against the new ModelSheet authoring environment, which is based on a new symbiosis of visual layout and program structures. ModelSheet retains the key strengths of spreadsheets – supporting decentralized innovation by domain experts who may not be programmers – while it improves model reliability, model expressiveness, team collaboration, and modeler productivity.

Keywords: **business analytics; business intelligence; information technology (IT); spreadsheet authoring; ModelSheet**

INTRODUCTION

Spreadsheets and word processors were the first major "WYSIWYG" (What You See is What You Get) applications of computers. They drove the desktop revolution in business (1980-2000) and, along with video games in the home, the broad adoption of early personal computers. Spreadsheets succeeded because many business analysts prefer to author models represented visually as tables and sheet layouts, instead of as pages of program code. Spreadsheets empowered decentralized innovation in business analysis by people with limited programming backgrounds.

The key innovations of spreadsheets are (1) to represent the logic of models to authors with WYSYWIG sheet layouts, and (2) to express virtually all relationships with formulas that are subordinate to cell layout and that are expressed in terms of cell addresses.

In 1993, Microsoft Excel 5.0 introduced multiple worksheets and pivot tables. Since that time, improvements in spreadsheets included performance, macros, pivot tables, algorithm libraries, model size limits, optional add-ons and other features.

By basing all models on cells, spreadsheets simplified modeling: the only basic concepts you need to master are sheet layout, cell addresses, and cell formulas. However, spreadsheets greatly proliferate cell formulas that relate cell addresses. Most quantitative models are more naturally expressed by relationships among variables: for example profit = revenue – expense, not C11 = D52 – Sheet2!E40. Spreadsheets don't even retain the concept of named variables, though named regions fulfill some of the functions of named variables in some situations.

Two competing concepts of simplicity are involved here. An analogy with computer programming languages might help.

- All programs can in principle be written in binary machine language as sequences of zeros and ones. Binary programming is simple because it is built from the fewest and simplest primitive operations.

- Higher level languages simplify the structure of programs rather than economizing on basic operations. They support iteration ("do"), conditional branching ("if"), subroutine calls, and so forth. These operations occur frequently in programs, and they map onto the way humans think about problems.

In the world of spreadsheets, building models from cells and cell formulas is analogous to programming computers in binary code; both technologies simplify the set of elementary operations. Programming with variables and other higher-level structures in ModelSheet is analogous to programming computers with higher level languages; both technologies simplify programs and the authoring process.

LIMITATIONS OF SPREADSHEETS

Proliferation of cell-level operations is the root cause of the four main weaknesses of spreadsheets.

- Reliability: Controlled experiments with spreadsheets indicates that spreadsheet models have serious reliability issues.[Powell et al (2007)]
  - In three experiments on entering information into cells, the percentage of cells with errors ranged from 11.3% to 21%. [Panko, R (1998); Panko, R. (2006); Panko, R. et al (1996)]
  - In one type of experiment, subjects are given a word problem and asked to create an appropriate spreadsheet model. Cell error rates ranged from 2% to 17%. The percentage of spreadsheets with at least one error ranged from 24% to 86%.
  - Audits since 1995 of spreadsheets deployed in field applications yield estimates that about 94% of field-deployed spreadsheets contain errors and cell error rates average 5.2%.
  - Another experiment tested auditing of spreadsheets that had errors planted in them. Subjects who had access to the numbers and formulas found no more errors than those who could check only the numbers.
- Expressiveness: The difficulty of changing model logic or layout often tips the cost-benefit tradeoff in favor of "satisficing" with a less realistic model.
- Team Collaboration: People commonly collaborate in providing data for spreadsheet models and in using spreadsheet reports. It is much less common for people to collaborate on building the formulas of a spreadsheet model. The basic reason for this is that collaboration occurs at the level of cells.
- Modeler Productivity: Productivity of modelers is reduced by difficulty interpreting cell addresses and cell formulas, proliferation of manual cell-level operations, the interdependence of model logic and layout.

As a result of these limitations, the focus of innovation in business modeling has moved from spreadsheets to business intelligence (BI) systems and enterprise resource planning (ERP) systems. This shift in innovation is evident in the revenue and revenue growth of the two product categories. BI and ERP generate $40-50 billion in revenue growing at 7-10% per year, and they are expected to continue to grow at these rates for

years. [Feld, C. (2007] Total revenue from spreadsheets is $3-4 billion and is probably growing at less than 10% per year.

BI and ERP undo some of the benefits of the desktop revolution. They strengthen centralized quality control over data and analyses, at the expense of limiting decentralized ad-hoc innovation. While they employ visual layout in some aspects of the authoring process, they revert to older paradigms of programming with code that makes modeling more difficult for domain experts who are not programmers.

These limitations of conventional spreadsheets open the door to a new spreadsheet authoring technology that strikes a better balance between visual sheet layout and programming structures such as variables, formulas, dimensions and time series.

Many companies regularly maintained spreadsheet reports and analyses that are used in reporting and decision support. A recent research report [Eckerson et al (2008)] defines a spreadmart as

> a reporting or analysis system running on a desktop database (e.g., spreadsheet, Access database, or dashboard) that is created and maintained by an individual or group that performs all the tasks normally done by a data mart or data warehouse, such as extracting, transforming, and formatting data as well as defining metrics, submitting queries, and formatting and publishing reports to others.

Highlights from the report:

- Over 90% of organizations have an average of 20-30 spreadmarts. Spreadmarts are usually created by analysts who prefer direct control of reports and analyses over reliance on the I.T. organization.
- Spreadmarts are at their best as one-off reports and ad-hoc analyses. Key benefits are faster deployment, faster response to change, more local control, ease of use, easier prototyping of a permanent system, more realistic capture of business logic.
- The main risks of reliance on spreadmarts (compared to centralized I.T.-managed solutions) are inconsistent views of data, large time commitment by analysts and higher costs, lack of audit trail.
- I.T.-managed reporting and analysis systems provide better data integrity and consistency; free business users to do their main jobs; better documentation, auditing and stability; and better scalability to larger user groups.
- 80% of affected analysts and 60% of affected managers view spreadmarts as a good solution; 80% of I.T. professionals want to eliminate them; executives are caught in the middle. Executive mandates to eliminate spreadmarts or stop I.T. support for them achieve the desired results only 6% of the time.

DESIGN CRITERIA FOR AN AUTHORING ENVIRONMENT FOR BUSINESS MODELS

In order to solve these spreadsheet problems, we would like an authoring environment that meets these six design criteria.

1) *Capture model logic with named symbolic variables and formulas that tell you in English what they mean*.
2) *Capture aspects of model structure with reusable dimensions and time series*. Spreadsheets capture these relationships with contiguous layouts of cells.
3) *Allow formulas to apply to an entire variable, a cell, or a scope in between.*
4) *Automate or eliminate most of the manual cell-level operations in modeling and layout*.
5) *Separate model logic and sheet layout*. The modeler can specify model logic without affecting sheet layouts, and he can specify layout without affecting model logic.
6) *Generate conventional spreadsheet files as output reports*.

MODELSHEET – A SOLUTION

ModelSheet is an authoring environment for business models that meets all six design criteria listed above. It delivers four primary benefits that improve on conventional spreadsheets.

1) Improved reliability of models.
2) Improved model expressiveness. It is more feasible to build more realistic and complex models.
3) Improved team collaboration. Teams collaborate at the level of concepts and variables, not cells.
4) Improved modeler productivity.

The figure below explains how the key features of ModelSheet generate these benefits.

Figure 1: How Features of ModelSheet Deliver Benefits

Innovative Features	Key Benefits			
	Model Reliability	**Model Expressiveness**	**Team Collaboration**	**Modeler Productivity**
Named symbolic variables Fewer formulas Readable formulas	• Eliminates misinterpretation of cell addresses • Reduces errors in writing & interpreting formulas • Facilitates auditing	• Easier to manage complex models with named variables, fewer formulas, more readable formulas.	• Collaborate at level of named variables, fewer and more readable formulas, not cell addresses.	• Grasp meaning of variables and formulas faster. • Fewer manual operations to enter formulas.
Reusable Dimensions and Time Series Attach formulas to nodes at any level, not just cells.	• Eliminates unnecessary repetition of manual tasks, hence reduces errors.	• Easier to include large, numerous, or hierarchical dimensions in models. • Easier to include e.g. annual sums in quarterly reports.	• Easier to edit someone else's work in one place than in each instance of a Dim or Time Series.	• Eliminates unnecessary repetition of manual tasks, hence saves time.
Separates model logic, sheet layout	• Reduces logic errors during modeling due to not having to consider sheet layout. • Eliminates logic errors caused by changing sheet layouts.	• Easier to author/edit complex models without worrying about layout. • Easier to author/edit layout without worrying about model logic.	• Easier to edit some else's model without worrying about interaction of model logic and layout.	• Authoring/editing complex model logic is faster without worrying about layout. • Authoring/editing complex layouts is faster without worrying about model logic.

Innovative Features	Key Benefits			
	Model Reliability	**Model Expressiveness**	**Team Collaboration**	**Modeler Productivity**
Automates manual cell-level operations	• Reduction on manual operations reduces errors.	• Easier to build complex models due to reduction of number of manual operations.	• Easier to edit some else's work due to fewer manual operations.	• Reduction on manual operations saves time.
Auto-generates spreadsheets Re-imports edits from spreadsheets	• No errors re-importing edited data from spreadsheet reports.	• Easier to incorporate new data assumptions in complex models.	• Easier to incorporate edited data from team members.	• Push one button to generate Excel workbook. • Automates listing and importing of new data assumptions.

BASIC ELEMENTS IN MODELSHEET MODELS

Spreadsheet users are already familiar with the four basic elements in ModelSheet models: Analysis Variables, Dimensions, Time Series, and Workbooks/Worksheets. Analysis Variables are the heart of a ModelSheet model.

Dimensions and Time Series

A dimension segments the values of variables. For more information see also http://en.wikipedia.org/wiki/Dimension_%28data_warehouse%29.

A time series specifies the time dependence of variables. It consists of a time range (a starting time and ending time) and a time grain (a basic time period such as month or year).

You can re-use a Dimension or Time Series in many Analysis Variables. You can edit once and all affected variables change. Dimensions and Time Series together define the table layout of an Analysis Variable, thereby freeing tables from being defined by cell addresses.

Dimensions and Time Series can have hierarchy. For example, a hierarchical Dimension of countries can have a global total, hemisphere, continents, countries, states within countries, and more.

ModelSheet computes the roll-up values for each Analysis Variable automatically, once you specify the "Accounting Type" (or other roll-up properties) of the variable. Here are some examples of how different Accounting Types translate the concept of "roll-up" into different arithmetic operations.

- For revenues or expenses, ModelSheet rolls up product values by adding them.
- For prices, ModelSheet has two methods for rolling up over a dimension.
  - Compute the unweighted average of the product prices.
  - The modeler specifies a roll-up formula, such as Price = Revenue/Sales_Units. This method yields the correct average selling price over the product line, if the modeler provides the required data.

Similarly, Time Series translate roll-up over Time Periods into different arithmetic operations. Examples:

- Revenue and expense roll up over time by addition.

- Assets, liabilities and equity normally roll up over time by taking the last value. This assumes these quantities are measured at the end of each time period. (There is also an Accounting Type for starting values.)
- Prices roll up over time with the same two methods as rolling up over Dimensions: unweighted averaging and a formula supplied by the modeler (such as Price = Revenue/Sales_Units).
- Growth or interest rates roll up over time by compounding.

Analysis Variables

Analysis Variables are the heart of a ModelSheet model. Typically there is one Analysis Variable for each key concept in the model (for example profit, revenue and expense). Spreadsheet users are familiar with Analysis Variables as tables of contiguous cells with the same conceptual meaning. However, conventional spreadsheets do not define variables as structures that can be used in formulas.

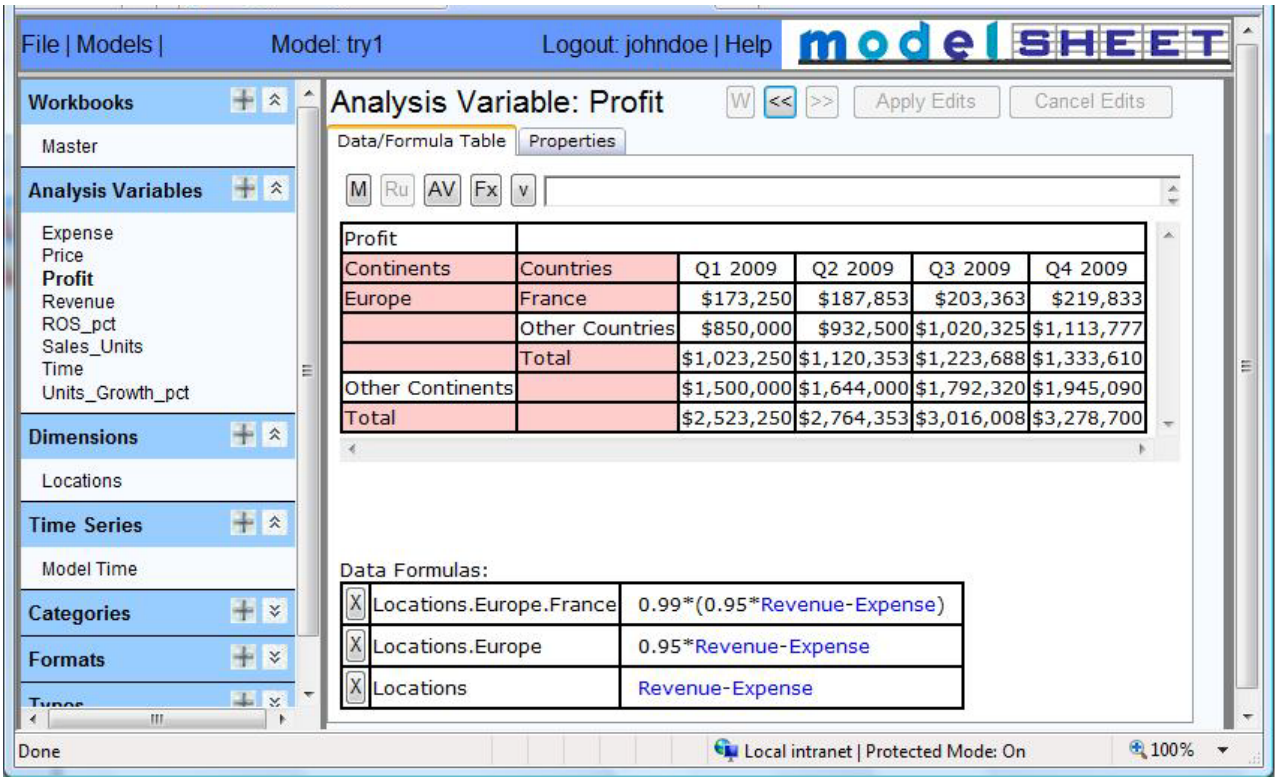
Analysis Variables in ModelSheet combine the key attributes associated with variables in a mathematical or programming model.

- The name of the variable (ideally) clearly communicates the meaning of the values in the variable.
- Dimensions and Time Series specify how to segment an Analysis Variables by dimensions and time.
- Each data value or formula specifies the value contained in a cell or region of cells in the variable table.

The following example illustrates these ideas by computing profits with different taxes in different countries.

- The formula Profit = Revenue – Expense defines profit globally in all time periods; it is associated with the "Total" node in the variable table.
- A 5% sales tax in Europe for all time periods might be represented by a formula Profit = 0.95*Revenue – Expense that is associated with the node for Europe in the Locations Dimension. This formula overrides the global formula in Europe.
- An additional 1% income tax surcharge in France might be represented by a formula Profit = 0.99*(0.95*Revenue – Expense) that is associated with the node for France. This formula overrides the European formula in France.

Figure 2: Formulas of Narrower Scope Override Formulas of Broader Scope



This use of Dimensions and Time Series, and precedence rules for formulas of different scopes, supports two beneficial features of ModelSheet:

- ModelSheet models usually have fewer formulas than conventional spreadsheets.
- Visual layout helps modelers to keep track of the scope of formulas in Analysis Variables. Most spreadsheet modelers usually prefer to use visual layout where possible to understand models.

In ModelSheet, variables, formulas, and cells have relationships that differ from those in conventional spreadsheets.

- Analysis Variables contain cells and formulas. The formulas define relationships among Analysis Variables. You can get lists of precedent and dependent variables of a given variable.
- In conventional spreadsheets, cells contain formulas, and cell layouts define variables. Formulas are expressed in terms of cell addresses and they define relationships among cells. You get lists of precedent and dependent cells.

Workbooks/Worksheets

ModelSheet has web workbooks that contain web worksheets. At any time, the author can push a button and generate Excel workbooks that have the same worksheets, layout and cell logic as ModelSheet's web workbooks.

The most basic difference from conventional spreadsheets is that ModelSheet completely separates model logic from sheet layout.

- In ModelSheet, relationships among the Analysis Variables are determined by symbolic formulas. Only when you ask for an Excel workbook does ModelSheet translate these relationships into cell formulas.
- In conventional spreadsheets, the relationships among cells are determined by cell formulas that are expressed in terms of cell addresses. This method intertwines model logic and sheet layout.

Separating model logic from sheet layout improves the entire authoring process. When you are editing model logic, you don't have to consider sheet layout; and when you are editing sheet layout, you can't accidentally or intentionally change model logic.

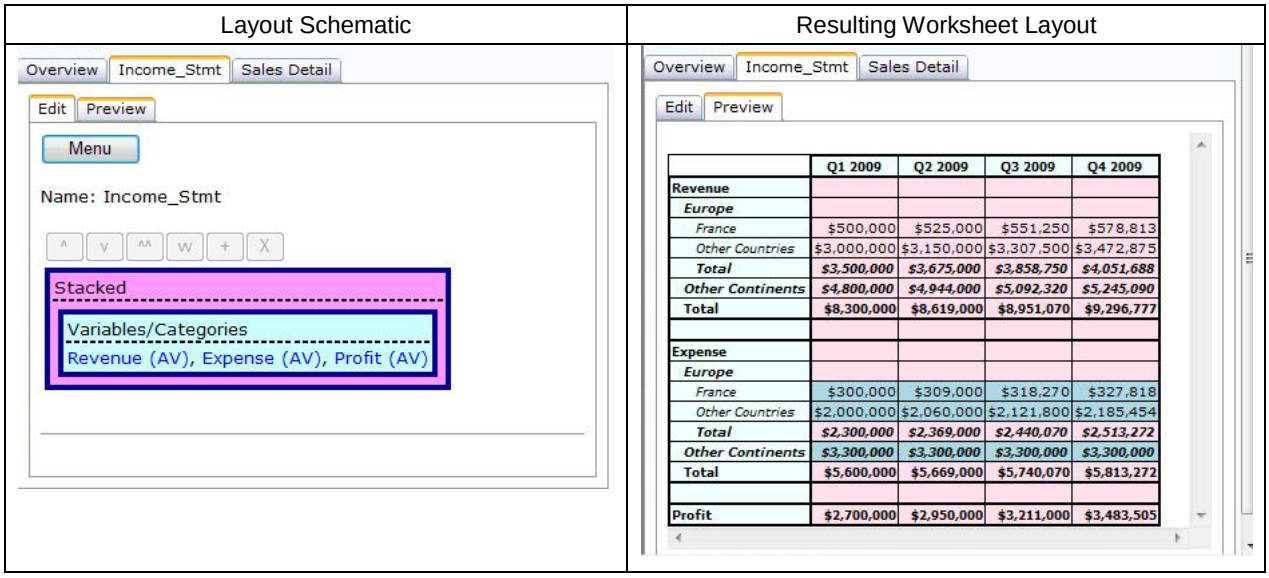
ModelSheet retains the advantages of using visual layout during authoring in two ways: (1) it retains the table layout for each Analysis Variable; and (2) worksheet layouts convey groupings of variables, as they do in conventional spreadsheets.

Modelers specify worksheet layouts in broad terms, such as which Analysis Variables to include, which variables are above/below or left/right of others, titles, and title formats. ModelSheet automatically specifies most of the remaining details of sheet layout. You can specify more layout details if you choose, such as the number or rows or columns that separate variables, tables and titles from one another.

When you add, delete or resize Analysis Variables, sheet layouts automatically change to accommodate the new information. You can make further manual adjustments to the layout if you choose.

The figure below shows the layout schematic (or "cartoon") that the modeler used to lay out revenue expenses profit in a simple income statement.

Figure 3: Worksheet Layout Schematic and resulting layout of a simple income statement

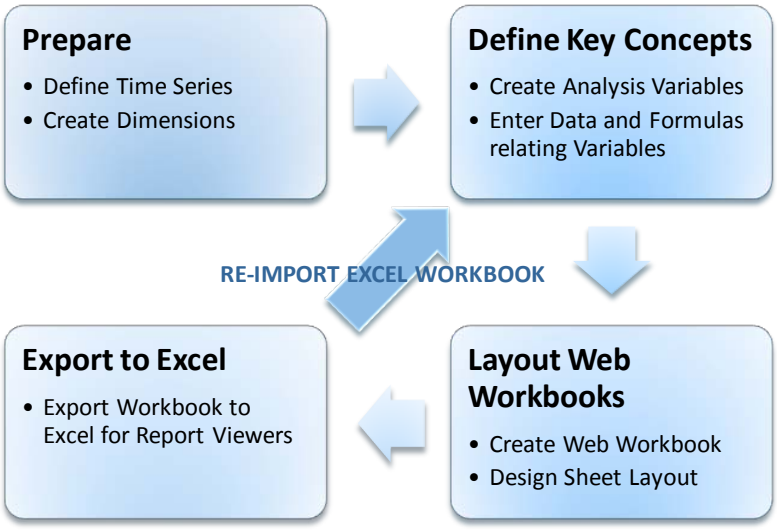


THE MODELING WORKFLOW

The structural differences between ModelSheet and conventional spreadsheets cause differences in the modeling workflow. We recommend performing the main steps in the order indicated below.

- Build supporting structures like Time Series and Dimensions.
- Define Analysis Variables, typically one for each key concept in the model. Inside each Analysis Variable, enter the formulas that determine cell values.
- Define workbooks and worksheets. Lay out worksheets.
- Generate Excel workbook(s) from the model and distribute to report users.
- Re-import changes in numerical inputs that report users enter in generated Excel workbooks.

Figure 4: The ModelSheet Workflow



We recommend the reader view the videos at http://www.modelsheetsoft.com/videos.aspx. They cover editing models, building a Model from Scratch, managing time series, and other topics.

SAMPLE APPLICATIONS

Readers who have ModelSheet user accounts can explore the prebuilt models. Readers who do not have a ModelSheet account can explore Excel workbooks generated from the ModelSheet models at http://www.modelsheetsoft.com/apps.aspx or http://office.microsoft.com/en-us/results.aspx?qu=modelsheet. However, the advantages of ModelSheet – such as named symbolic variables, readable formulas and far fewer formulas, automation of manual cell-level operations – have been removed in the process of translating the models to Excel workbooks.

We recommend examining the following prebuilt models.

- Business Unit Financial Plan
- Investment Project Financial Analysis
- Operations Process Flow Analysis
- Product Profitability Model
- Marketing Program Contribution Margins

Conclusion

The conventional spreadsheet paradigm, though enhanced, has remained essentially the same since the late 1970s. The longevity of the design is a tribute to the strength of the spreadsheet concept. Spreadsheets are excellent for a wide range of reporting tasks. However, conventional spreadsheets are inadequate for authoring the larger and more complex applications for which spreadsheets are now used.

To fix these problems, we listed six design criteria needed to design an authoring environment. ModelSheet satisfies these six design principles and, we believe, delivers corresponding improvements in reliability, expressiveness, collaboration, and productivity.

For more information, visit http://www.modelsheetsoft.com.

REFERENCES

Powell, Stephen G., Baker Kenneth R., & Lawson, Barry, A (2007), Critical Review of the Literature on Spreadsheet Errors, Tuck School of Business, Dartmouth College.

Panko, R (1998), What we know about spreadsheet errors, *Journal of End-User Computing* 10, 15-21.

Panko, R. (2006), What we know about spreadsheet errors, http://panko.cba.hawaii.edu/ssr/Mypapers/whatknow.htm.

Panko, R., Halverson, R. (1996), Spreadsheets on trial: a survey of research on spreadsheet risks, *Proceedings of the 29th Annual Hawaii International Conference on Systems Sciences*, 326-335

Abbott, J. (2001). Data data everywhere - and not a byte of use? *Qualitative Market Research, 4 (3),* 182-192.

Abbott, J., Stone, M., & Buttle, F. (2001). Customer relationship management in practice: A qualitative study. Journal of Database Marketing, 9 (1), 24-34.