The ModelSheet® Authoring Environment for Spreadsheet Models

Richard J. Petti and Howard I. Cannon
ModelSheet Software, LLC
E-mail: info@modelsheetsoft.com
URL: http://www.modelsheetsoft.com
Blog: http://blog.modelsheetsoft.com
Group: http://groups.yahoo.com/group/modelsheet

July 20, 2008, revised through March 24, 2011

Abstract: ModelSheet Authoring is an environment for building and editing business models that exports models as conventional spreadsheets. It addresses the main weaknesses of spreadsheets while retaining a role for visual sheet layout during authoring. ModelSheet Authoring introduces six key design features. (1) It captures model logic with named variables and symbolic formulas that tell you in natural language what they mean. (2) It defines dimensions and time series that can be reused in many variables and centrally edited, instead of relying on contiguous cell layout to capture these structures. (3) ModelSheet attaches formulas to dimension nodes and time periods, propagates formulas from higher level nodes to lower level nodes, and applies precedence rules when formula scopes overlap. Therefore ModelSheet models usually have 1%-10% as many formulas as conventional spreadsheets. (4) ModelSheet Authoring automates virtually all the manual cell-level operations in specifying model logic and sheet layout. (5) ModelSheet completely separates authoring of model logic from authoring of sheet layout. (6) ModelSheet Authoring exports customizable spreadsheet models and/or fully functional conventional spreadsheet workbooks as output. These features enable ModelSheet to improve model expressiveness, model reliability, team collaboration, and modeler productivity.

ModelSheet Authoring is the technology engine behind ModelSheet’s Customizable Spreadsheets.

Contents

1 Introduction 2
2 ModelSheet’s Solution 5
3 Basic Elements in ModelSheet Models 7
4 ModelSheet Authoring in Action 14
5 Conclusion 17

1 This paper: http://templates.modelsheetsoft.com/my/getfile.aspx?name1=whitepaper-Authoring-intro
2 "Customizable Spreadsheets – The Promise of Templates Fulfilled", by Richard J. Petti and Howard I. Cannon,
1 Introduction

1.1 Spreadsheets: a Landmark Innovation in Modeling and Analysis

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 they empower business analysts prefer to author models represented visually as tables and sheet layouts, instead of as pages of program code. They enabled decentralization of business analysis, and they empowered many people with limited programming background to build their own models and to innovate without excessive dependence on programmers.

The key design innovations of spreadsheets are (1) to represent the logic of quantitative models with WYSIWYG sheet layouts during the authoring process, and (2) to express virtually all relationships with formulas that are subordinate to cell layout³ and that are expressed in terms of cell addresses.

Innovation continued. Lotus Improv pioneered pivot tables, but in a product separate from its spreadsheet cousin, Lotus 1-2-3. In 1993, Microsoft Excel 5.0 introduced multiple worksheets and pivot tables and VBA macros. Since that time, many substantial improvements were made in Excel: performance, algorithm libraries, model size limits, named regions, array functions, optional add-ins and other features. However, the basic modeling paradigm of spreadsheets has not changed since the late 1970s – to capture model logic with sheet layout, and to express relationships at the cell level.

Spreadsheets have been so successful compared to other modeling approaches that they are now applied to models that far exceed in size and complexity the problems they were designed to handle.

For more background on spreadsheets, see http://en.wikipedia.org/wiki/Spreadsheets.

1.2 Two Kinds of Simplicity

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 expressed with cell addresses.⁴ Most quantitative models are more naturally and easily 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 and array functions 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 illustrate the differences between them.

• All computer 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 most basic elements possible.

• Higher level languages simplify the structure of programs rather than economizing on basic operations. They support iteration (“do loops”), conditional branching (“if statements”), subroutine calls, and so forth as part of the language. These operations occur frequently in programs, and they map onto the way humans think about computing problems.

The computer revolution would long ago have slowed to a crawl if most programs were authored in binary code. For the overwhelming majority of application programs, simplifying programs and the authoring

³ that is, each formula is typically assigned to one cell in the sheet layout.

⁴ Conventional programming languages would use a smaller number of array variables that contain and organize the array elements.
process is far more important than reducing the number of elementary operations in the authoring environment.

In the world of spreadsheets, building models from cells and cell formulas is analogous to programming computers in binary code; both technologies have minimal primitive elements and basic 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.

1.3 Limitations of Spreadsheets

The limited range of modeling structures, with consequent proliferation of cell-level operations, is the root cause of the main weaknesses of spreadsheets. These weaknesses are of two types: weaknesses in the models themselves, and organizational implications.

1.3.1 Weaknesses of Spreadsheet Models

- **Management of Intellectual Property (IP):** Spreadsheets store IP in an executable form, but the excessively large number of inscrutable cell formulas impairs the optimal use of IP stored in spreadsheets. IP is often hard to access, hard to communicate, and hard to exploit.

- **Model Expressiveness:** The difficulty of changing model logic or layout often tips the cost-benefit trade-off in favor of “satisficing” with a less realistic model. Cost-benefit tradeoffs abound in life, so this situation should not be surprising. The key point is that conventional spreadsheets offer a suboptimal tradeoff between costs and benefits of improvements in moderately complex models, compared to what is possible with ModelSheet.

Despite these problems, expressiveness in spreadsheet models turns out to be superior to that of programming languages in some practical situations. Some financial services organizations have told the authors that they use spreadsheets for mission-critical work because the alternative – that expert analysts communicate their design and performance criteria to professional programmers – cannot faithfully and efficiently handle all the subtle modeling criteria known to the analyst.

- **Model Flexibility:** One a spreadsheet model is built (we have been told by numerous heavy-duty spreadsheet users), making moderately complex edits is often so difficult that it is easier to use the existing spreadsheet as a model and start over to implement a new version. This problem is particularly acute if the original author is not involved, or the author has not reviewed the model for a few years.

- **Reliability:** Controlled experiments with spreadsheets indicates that spreadsheet models have serious reliability issues.\(^5\)\(^6\)
  - In three experiments on entering information into cells, the percentage of cells with errors ranged from 11.3% to 21%.\(^7\)\(^8\)\(^9\)
  - 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%.

---


\(^6\) One financial services firm told the authors they have reduced reliability problems of spreadsheets to levels of other modeling technologies. We believe that this was accomplished largely by career-long selective pressures for specialized skills and by very high compensation levels.


– 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 in the spreadsheets found no more errors than those who could check only the numbers. While visual layout can help in auditing spreadsheets, the hard-to-interpret cell formulas are of little help.

1.3.2 Organizational Impact

• **Turnaround Time:** Turnaround times for building spreadsheets are sometimes faster than other technologies, when you factor into communications between domain experts and programmers for competing technologies. However, turnaround times for making modest edits to spreadsheets are sometimes extremely long. Example: Consider a spreadsheet model with a product line to which you want to add a new product, or reorganize the products into product families. If the product line appears on a few dozen tables, these changes are time consuming and error-prone. Productivity of spreadsheet modelers is reduced by the difficulty in interpreting cell addresses and cell formulas, the proliferation of manual cell-level operations, the interdependence of model logic and layout, and the absence of higher level structures like variables, dimensions, and time series.

• **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 this sort of collaboration must occur at the level of cells that contain inscrutable cell formulas. The meanings of cell addresses and cell formulas are relatively obscure even to the author of a model, if the model is complex, even to the author if he has not worked with a model for a few months.

• **Employee Turnover:** Many managers of activities that depend on spreadsheets live in fear of employee turnover and succession, because a new person must learn the model. Turnover of supervisors can be even more difficult than turnover of authors, because highly skilled supervisors often manage the best practices that make spreadsheet problems more manageable.\(^\text{10}\)

• **Auditing:** Spreadsheets are particularly difficult to audit because the number of formulas is much greater than the conceptual complexity of a typical model requires, and because each formula is in inscrutable cell formula, not a readable symbolic formula.

As a result of these limitations and the size limitations of spreadsheets, much of the 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.\(^\text{11}\) Total revenue from spreadsheets is $3-4 billion and is probably growing at less than 5% per year.

However, BI and ERP undo some of the benefits of the desktop revolution. They strengthen centralized quality control of 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.

1.4 The Impact of Spreadsheets on Information Use in Organizations

Many companies regularly maintained spreadsheet reports and analyses that are used in reporting and decision support. In a recent research report, The Data Warehousing Institute (TDWI) 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.

The TDWI report summarizes the benefits and problems of spreadmarts. Our digest of their key findings follows.

- Over 90% of organizations have an average of 20-30 spreadmarts.
- Spreadmarts are usually created by analysts in finance, marketing, and other departments, or in financial services firms, who prefer direct control of the reports and analyses over reliance on the I.T. organization.
- Key benefits of spreadmarts are faster deployment, faster response to change, more local control by the user community, ease of use (familiarity of users with spreadsheets), lower cost, easier customization, easier prototyping of a permanent system, more realistic capture of business logic, compared to systems produced by I.T. organizations.
- 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 centrally manage data and business logic in a documented process that users can access. They provide better data integrity and consistency; more freedom of business users to do their main jobs (which are not to develop models); better documentation, auditing and stability; and better scalability to larger user groups.
- Spreadmarts are at their best as one-off reports and ad-hoc analyses. The problems they cause are greatest when spreadmarts are used as regular business systems.
- 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. Spreadmarts satisfy real needs that are hard for I.T. organizations to satisfy.

2 ModelSheet’s Solution

ModelSheet Authoring aims to amplify the advantages and reduce the problems of spreadmarts by replacing spreadsheets with a superior authoring environment, while strengthening spreadsheets as report delivery medium.

Before delving into details, this section compares conventional spreadsheet authoring with ModelSheet Authoring at a conceptual level.

---

2.1 Key Features of ModelSheet Authoring

Six principal features account for the benefits of ModelSheet.

1) **ModelSheet Authoring captures model logic with named symbolic variables and formulas that tell you in English what they mean.** ModelSheet’s variables contain formulas that express relationships among variables. A ModelSheet formula can apply to an entire variable, an individual cell, or a scope in between.

2) **ModelSheet Authoring captures aspects of model structure with reusable dimensions and time series.** You edit a dimension or time series once and the changes will be automatically included in the appropriate variables, and layouts will be automatically adjusted to accommodate changed sizes of tables. Spreadsheets capture these relationships with contiguous layouts of cells.

3) **ModelSheet Authoring attaches formulas to dimension nodes and time periods.** It propagates formulas from higher level nodes and time periods to lower level nodes and time periods, and applies precedence rules when formula scopes overlap. Therefore ModelSheet models typically have 1%-10% as many formulas as conventional spreadsheets.

4) **ModelSheet Authoring automates virtually all the manual cell-level operations in modeling and layout.** ModelSheet assigns ‘accounting types’ to variables to facilitate automation of roll-up operations over time and dimensions. For example, it knows how to combine profit, profit %, assets, and growth rates over countries and over time periods with minimal manual intervention. By contrast, conventional spreadsheets require that authors compose roll-up formulas manually, and copy and adapt them (such as by fixing components of cell addresses with dollar signs) to the environment of each cell.

5) **ModelSheet Authoring completely separates model logic and sheet layout.** You can specify model logic including variables, dimensions and time series without specifying anything about sheet layouts. Conversely, you can be sure that any change you make in sheet layout will not change the model logic. By contrast, spreadsheet logic is inextricably intertwined with sheet layout through reliance on cell addresses and cell formulas. This is a major cause of “spreadsheet hell” and low reliability.

6) **ModelSheet Authoring exports conventional spreadsheet files as output reports.** Modelers work in a powerful authoring environment, and report users get fully functional conventional spreadsheets complete with cell-based formulas. ModelSheet also enables you to re-import edited data from exported Excel workbooks. The useful contrast here is not with spreadsheets, but with vertical analysis software that tries to displace spreadsheets with specialized reporting software.

2.2 Benefits of ModelSheet Authoring

ModelSheet Authoring delivers many benefits that improve on conventional spreadsheets and authoring processes.

1) ModelSheet Authoring improves management of intellectual property.

2) ModelSheet Authoring improves model expressiveness. It is more feasible to build more realistic and complex models.

3) ModelSheet Authoring improves model flexibility by making it much easier to modify models after they are built.

4) ModelSheet Authoring improves turnaround time for building and editing spreadsheet models.

5) ModelSheet Authoring improves team collaboration and the impact of staff turnover. Teams collaborate at the level of concepts and variables, not cells.

---

13 Reduced to basics, ModelSheet Authoring builds models from variables that contain formulas and cells, whereas conventional spreadsheets build models from cells that contain formulas and there are no variables. Sometimes the dependence of spreadsheet formulas on cell addresses can be mitigated by using named regions, but this is an inadequate solution.
6) ModelSheet Authoring improves reliability of models and makes them easier to audit.

The figure below attempts to compress the many factors that explain how the features of ModelSheet Authoring deliver these benefits.

**Figure 1: How ModelSheet Authoring Delivers Some of Its Key Benefits**

<table>
<thead>
<tr>
<th>Innovative Features</th>
<th>Key Benefits</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Intellectual Property</strong></td>
<td><strong>Model Expressiveness</strong></td>
</tr>
<tr>
<td>Named symbolic variables</td>
<td>Fewer readable formulas make IP more accessible.</td>
</tr>
<tr>
<td>Fewer formulas</td>
<td>Time series and dimensions make IP faster and easier to deploy in new situations.</td>
</tr>
<tr>
<td>Readable formulas</td>
<td>IP can be express in compact set of readable formulas without reference to sheet layouts.</td>
</tr>
<tr>
<td>Reusable dimensions and time series</td>
<td>Attach formulas to nodes at any level, not just cells.</td>
</tr>
<tr>
<td>Separates model logic, sheet layout</td>
<td>Automates manual cell-level operations</td>
</tr>
<tr>
<td>Automates manual cell-level operations</td>
<td>Exports spreadsheets Re-imports edits in exported spreadsheets</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### 3 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. However, we usually recommend specifying dimensions and time series before analysis variables so you can use them when specifying analysis variables. So we will start with dimensions and time series.
3.1 Dimensions and Time Series

A dimension segments the values of variables. In the figure above, Revenue and COGS (cost of goods) are segmented by products, and the two products form a dimension.\footnote{For more information see also \url{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). In the figure above, the three months January through March specify a time range and a time grain, so they specify a time series.\footnote{You can think of a time series as a special kind of dimension for which you can do interval arithmetic; that is, you can subtract two dates and get a number (a number of days in ModelSheet).}

Dimensions and time series are common in spreadsheet models. However, in conventional spreadsheets, a dimension or time series is merely a list of similar items laid out in contiguous cells in a row or column. Spreadsheets have no data structures corresponding to dimensions or time series, except in pivot tables, where the "fields" are rudimentary dimensions with no hierarchy.

In ModelSheet Authoring you explicitly create structures called dimensions and time series, for two reasons. First of all, you can re-use a dimension or time series in many analysis variables. Secondly, dimensions and time series together define the table layout of an analysis variable, and thereby define the cells in the table (instead of defining the table with cell addresses and defining dimensions and time series by contiguous layouts of cells). Freeing dimensions, time series, and variable tables from dependence on
cell addresses is a critical part of separating model logic from sheet layout, which is a key advantage of ModelSheet.

3.1.1 Hierarchy in Dimensions and Time Series

Dimensions and time series can have hierarchy. For example, a hierarchical dimension of countries and a hierarchical time series are depicted in the figures below.

<table>
<thead>
<tr>
<th>Figure 3: A Hierarchical Dimension</th>
<th>Figure 4: A Time Series Hierarchy</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Dimension “Countries”</strong></td>
<td><strong>Time Series “Model Time”</strong></td>
</tr>
<tr>
<td>• Eastern Hemisphere</td>
<td>• 2011</td>
</tr>
<tr>
<td>– Africa</td>
<td>– Q1</td>
</tr>
<tr>
<td>Algeria</td>
<td>January</td>
</tr>
<tr>
<td>…</td>
<td>…</td>
</tr>
<tr>
<td>• Asia</td>
<td>• 2012</td>
</tr>
<tr>
<td>– Africa</td>
<td>– Q4</td>
</tr>
<tr>
<td>Algeria</td>
<td>October</td>
</tr>
<tr>
<td>…</td>
<td>…</td>
</tr>
<tr>
<td>• Western Hemisphere</td>
<td>• 2011</td>
</tr>
<tr>
<td>– North America</td>
<td>– Q1</td>
</tr>
<tr>
<td>Canada</td>
<td>January</td>
</tr>
<tr>
<td>…</td>
<td>…</td>
</tr>
<tr>
<td>• South America</td>
<td>• 2012</td>
</tr>
<tr>
<td>Brazil</td>
<td>– Q4</td>
</tr>
<tr>
<td>…</td>
<td>October</td>
</tr>
</tbody>
</table>

If this dimension segments an analysis variable by country, then ModelSheet Authoring’s default behavior is to roll up values for countries to show subtotals for each continent, to roll up values for continents to show subtotals for each hemisphere, and to roll up values for hemispheres to show a global total.

Similarly, if the time series above segments an analysis variable by time, then ModelSheet Authoring can roll up the months to show subtotals for quarters, and roll up the quarters to show subtotals for each year, and roll up the years to show a total for the entire time series. ModelSheet Authoring’s default behavior is to show values for only one time grain; you must choose to display values for two time grains if desired.

3.1.2 Roll-ups (Subtotals)

In Figure 2 above, Revenue and COGS roll up values for the two products and report total values. ModelSheet Authoring 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 an analysis variable that contains revenue or expenses, ModelSheet Authoring rolls up product values by adding them.
- For an analysis variable that contains prices, ModelSheet Authoring has two methods for rolling up product prices.
  - The first method is to compute the unweighted average of the product prices. This is the neutral or ‘vanilla’ method that ModelSheet Authoring can implement without any other information from the modeler.
The second method is to allow the modeler to specify a roll-up formula, such as \( \text{Price} = \frac{\text{Revenue}}{\text{Sales}_\text{Units}} \), or \( \text{Price} = \text{if}(\text{Sales}_\text{Units}=0, \text{List}_\text{Price}, \frac{\text{Revenue}}{\text{Sales}_\text{Units}}) \). This method yields the correct average selling price over the product line, if the modeler provides the required data.

Time series also have roll-up operations. For example we can roll up values for January, February and March to get a value for the first quarter. Different Accounting Types translate roll-up over Time Periods into different arithmetic operations. Here are some examples.

- For an analysis variable that contains revenue or expenses, ModelSheet Authoring rolls up values over time by adding them.
- For an analysis variable that contains assets, liabilities or equity, ModelSheet Authoring rolls up values over time by taking the last value. This assumes the usual convention that these quantities are measured at the end of each time period, so that the closing assets for March are the same as the closing assets for the first quarter.\(^{16}\)
- For an analysis variable that contains prices, rolling up over time series can be done with the same two methods as rolling up over dimensions: unweighted averaging and a formula supplied by the modeler (such as \( \text{Price} = \frac{\text{Revenue}}{\text{Sales}_\text{Units}} \)).
- For an analysis variable that contains growth or interest rates, rolling up over time is accomplished by compounding the rates.

Modelers have the option to suppress the roll-up behavior for a dimension or time series.

### 3.2 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 already familiar with analysis variables as tables of values that share the same conceptual meaning. However, conventional spreadsheets define variables as tables formed by a contiguous layout of cells, and not as a structure that can be used as a variable in formulas.

#### 3.2.1 Attributes of Analysis Variables

Analysis variables in ModelSheet combine most of 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 are associated with an analysis variables to specify how to segment it by dimensions and time. Examples: Products dimension, Geographic Locations dimension, Model Time (a time series that applies to most variables in a model). An analysis variable can have any number of dimensions, but it can have only zero or one time series (for now). The dimensions and time series together define the cell layout of the table for the analysis variable.
- Each data value or formula specifies the value contained in a cell or region of cells in the variable table.

#### 3.2.2 Data and Formulas Define Cell Values Inside Analysis Variables

Data and formulas can be attached to any dimension node or cell in a variable table. Data and formulas of narrower scope (i.e. that apply to fewer dimension items and fewer time periods) take precedence over formulas of broader scope.

The following example illustrates these ideas by representing the impact on profits of different taxes in different countries.

---

\(^{16}\) For Analysis Variables that contain starting values in time periods, ModelSheet has an Accounting Type called “Stock Start” that rolls up over time by selecting the value for the first time period. For example, if you are rolling up months to get quarterly values, then the starting value for the first quarter is defines as the starting value for January.
• 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 cell for France. This formula overrides the European formula in France.

Figure 5: 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 Authoring:

• ModelSheet models usually have fewer formulas than conventional spreadsheets. This analysis variable contains three formulas whereas the analogous spreadsheet model would have 15.

• 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.

3.2.3 Relationships between Variables, Formulas, and Cells

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

• In ModelSheet, analysis variables contain cells and formulas. The formulas define relationships among analysis variables. ModelSheet provides 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. Conventional spreadsheets provide lists of precedent and dependent cells, not variables, because there are no variables, only cells and cell addresses.
Business models are most powerfully expressed as relationships among variables, (such as Profit = Revenue – Expense), not relationships between cells. These basic design differences have a large impact on the modeling experience.

3.3 Workbooks/Worksheets

3.3.1 Model Logic and Worksheet Layout

ModelSheet Authoring has web workbooks that contain web worksheets. At any time, usually when the model is completed, the author can push a button and export Excel workbooks that have the same worksheets, layout and cell logic as ModelSheet Authoring’s web workbooks.

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

- The relationships among the analysis variables in a ModelSheet model are determined by the symbolic formulas in the model. Only when you ask for an Excel workbook does ModelSheet Authoring translate these relationships into cell formulas based on cell addresses in order to create a conventional spreadsheet file.
- By comparison, the relationships among cells (not variables) in conventional spreadsheets are determined by cell formulas that are built of cell addresses. This method intimately intertwines the representation of 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 consider sheet layout; and when you are editing sheet layout, you can’t accidentally or intentionally change model logic. This change in process improves model reliability, expressiveness, and productivity.

ModelSheet Authoring 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.

3.3.2 Semi-Automatic Sheet Layout

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.

In conventional spreadsheets, the author must specify the cell addresses for each table and title. The main exception to this rule is pivot tables, where Excel chooses actual cell addresses for information based on the sizes of filter fields and their logical arrangement (for example, which fields are in the left or top header).

3.3.3 Automatic Layout Adjustment

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.

In conventional spreadsheets, when you add a new variable to a worksheet or change the size of a table, you must manually adjust the layout at the cell level to avoid collisions of tables and other layout problems.

3.3.4 Examples of Web Workbooks

The figure below shows the layout schematic (or “cartoon”) that the modeler used to lay out revenue expenses profit in a simple income statement.
**Introduction to ModelSheet Authoring**

### Figure 6: Worksheet Layout Schematic and resulting layout of a simple income statement

<table>
<thead>
<tr>
<th>Layout Schematic</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Overview</strong></td>
</tr>
<tr>
<td><strong>Income Stmt</strong></td>
</tr>
<tr>
<td><strong>Sales Detail</strong></td>
</tr>
<tr>
<td><strong>Edit</strong></td>
</tr>
<tr>
<td><strong>Preview</strong></td>
</tr>
<tr>
<td><strong>Menu</strong></td>
</tr>
<tr>
<td><strong>Name:</strong> Income Stmt</td>
</tr>
</tbody>
</table>

**Stacked**

- Variables/Categories
  - Revenue (AV), Expense (AV), Profit (AV)

---

<table>
<thead>
<tr>
<th>Resulting Worksheet Layout</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Overview</strong></td>
</tr>
<tr>
<td><strong>Income Stmt</strong></td>
</tr>
<tr>
<td><strong>Sales Detail</strong></td>
</tr>
<tr>
<td><strong>Edit</strong></td>
</tr>
<tr>
<td><strong>Preview</strong></td>
</tr>
</tbody>
</table>

**Revenue**
- Europe
  - France: $800,000, $525,000, $551,250, $578,813
  - Other Countries: $3,000,000, $1,390,000, $2,347,873
  - Total: $2,000,000, $2,678,000, $3,958,720, $4,051,669

**Expense**
- Europe
  - France: $300,000, $309,000, $318,770, $327,818
  - Other Countries: $300,000, $309,000, $318,770, $327,818
  - Total: $1,800,000, $2,060,000, $2,512,727

**Other Continents**
- $4,000,000, $4,440,000, $4,082,200, $4,245,680
- Total: $10,000,000, $10,000,000, $10,000,000, $10,000,000

**Profit**
- $2,700,000, $3,200,000, $2,422,500

### Figure 7: Worksheet Layout Schematic and resulting layout that places some variables to right of existing variables

<table>
<thead>
<tr>
<th>Layout Schematic</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Overview</strong></td>
</tr>
<tr>
<td><strong>Income Stmt</strong></td>
</tr>
<tr>
<td><strong>Sales Detail</strong></td>
</tr>
<tr>
<td><strong>Edit</strong></td>
</tr>
<tr>
<td><strong>Preview</strong></td>
</tr>
<tr>
<td><strong>Menu</strong></td>
</tr>
<tr>
<td><strong>Name:</strong> Income Stmt</td>
</tr>
</tbody>
</table>

**Stacked**

- Side by side
  - Variables/Categories
    - Revenue (AV), Expense (AV), Sales Units (AV)
  - Variables/Categories
    - Profit (AV)

---

<table>
<thead>
<tr>
<th>Resulting Worksheet Layout</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Overview</strong></td>
</tr>
<tr>
<td><strong>Income Stmt</strong></td>
</tr>
<tr>
<td><strong>Sales Detail</strong></td>
</tr>
<tr>
<td><strong>Edit</strong></td>
</tr>
<tr>
<td><strong>Preview</strong></td>
</tr>
</tbody>
</table>

**Revenue**
- Europe
  - France: $500,000, $525,000, $551,250, $578,813
  - Other Countries: $3,000,000, $1,390,000, $2,347,873
  - Total: $6,000,000, $2,500,000, $5,512,727

**Expense**
- Europe
  - France: $300,000, $309,000, $318,770, $327,818
  - Other Countries: $300,000, $309,000, $318,770, $327,818
  - Total: $1,800,000, $2,060,000, $2,512,727

**Other Continents**
- $4,000,000, $4,440,000, $4,082,200, $4,245,680
- Total: $10,000,000, $10,000,000, $10,000,000, $10,000,000

**Profit**
- $2,700,000, $3,200,000, $2,422,500

#### 3.4 Exported Excel Workbooks

ModelSheet Authoring can export a model or parts of a model to one or more Excel workbooks, with the push of a button. Report users get the familiar Excel environment. They have no way to distinguish an Excel workbook exported by ModelSheet Authoring from one that was manually constructed; except perhaps by stylized layouts and the “Formulas” worksheet. The Formulas worksheet shows all the symbolic formulas.
expressed with named variables in the ModelSheet model, to help users understand the model. The symbolic formulas are not executable in Excel.

### 3.5 Macros and VBA

ModelSheet Authoring does not now support macros or VBA. Adding this capability involves implementation, but not fundamental changes to the system. We just haven’t gotten around to it yet, and we will make this extension for a project that requires macros.

### 4 ModelSheet Authoring in Action

#### 4.1 The Modeling Workflow

The structural differences between ModelSheet Authoring and conventional spreadsheets cause the ModelSheet workflow to be different. We recommend performing the main steps in the order indicated below. The modeler can also break these steps into pieces, for example define some dimensions first and return later to define more.

- Build supporting structures like time series and dimensions. You will need them to fully specify analysis variables.
- Define analysis variables, typically one analysis variable for each key concept in the model. Inside each analysis variable, enter the formulas that determine the cell values of the variable, and that define the relationships between variables.
- Define workbooks and worksheets. Lay out worksheets with analysis variables, titles, text formats, and spacing. Building quantitative model logic is separate from specifying sheet layout.
- Export Excel workbook(s) from the model and distribute to report users.
- Re-import changes in numerical inputs that report users enter in exported Excel workbooks.

#### Figure 8: The ModelSheet Authoring Workflow

![ModelSheet Workflow Diagram]

#### 4.2 Video Demonstrations

A narrated video is worth thousands of words in explaining the actual use of the product. We recommend viewing the following videos (and new ones that appear from time to time) at [http://www.modelsheetsoft.com/videos.aspx](http://www.modelsheetsoft.com/videos.aspx).

- Editing Models with ModelSheet Authoring
This video demonstrates how to debug and refine a simple pre-existing financial model of a business unit. The largest benefits of using ModelSheet Authoring accrue when exploring, editing and debugging models that are already built.

This is true of all CAD (Computer Aided Design) applications – and ModelSheet is “CAD for spreadsheets”; it uses more structural information about models than conventional spreadsheets do, just as CAD applications use more structural information about engineering designs than mechanical drawing packages do.

• Building a Model From Scratch with ModelSheet Authoring
  This demonstration shows every gesture needed to build a simple income statement containing six variables, numerical data and formulas that express relationships among the variables.

• Managing Time in ModelSheet Authoring
  Demonstrates ModelSheet Authoring’s facilities for managing time in models: changing time range, time grain; adding annual sums; displaying time as calendar dates, absolute fiscal dates, or relative fiscal dates.

4.3 Sample Applications
Readers who have ModelSheet Authoring user accounts can explore the prebuilt models. Readers who do not have a ModelSheet account can explore Excel workbooks exported from the ModelSheet models at http://templates.modelsheetsoft.com/browser/browse.aspx. 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. You can examine either in a ModelSheet model in a ModelSheet user account, or Excel workbooks exported by ModelSheet from the models. You can download a free working Excel workbook derived from each of the models described below. Each of the links below goes to a web page where you can get customizable spreadsheet templates built from the corresponding ModelSheet model.

4.3.1 Financial Analysis
  This model is a prebuilt financial plan with financial statements, ratio analysis, and valuation model. The revenue model includes list prices, actual prices, sales units, growth rates, sales returns. It tracks expenses by department, and has accounts for headcount, labor, facilities and other expenses. The balance sheet tracks several kinds of assets, liabilities, and equity. The advanced version has many additional optional features, such as recruiting expenses, untagged assets, leases, corporate bonds, capitalized development spending, production learning effects.

  Includes sources and uses of cash, entry of receivables and payables by account, and Excel databases for salaried and hourly employee expenses. The advanced version includes line of credit with asset borrowing base, optional Excel databases that track receivables and payables by invoice, and optional bank checking account cash float.
Computes investment, working capital, discounted cash flows, earnings, various measures of return, and valuation for an investment project. The model tracks several related sub-projects that each has its own investment amount, start date, depreciation method and life, working capital requirements and so forth. It has an optional CAPM formula for discount rate.

Tracks investments, common shares, and multiple series of preferred shares, options, and notes through multiple rounds of financing of a company. Tracks liquidiation preferences of senior securities and their effect on payouts for different valuations of the company. The model allows you to enter conversion and exercise decisions, and tries to estimate reasonable default decisions from its estimate of the prices of different securities.

Computes expense budgets for cost centers, based on activity counts for expense drivers and specified ratios (operating expense) / (activity count). The model estimates expense-per-activity ratios from historical data. It can handle multiple expense accounts per cost center, many expense accounts, many activity drivers, and numerous drivers per expense account.

4.3.2 Marketing and Sales Analysis

Combines managers’ input targets for revenue of selected market segments with trends extracted from historical data to forecast revenue and sales units in the future. Can combine managers’ targets for revenue of products, sales channels, customer industries, and total revenue in a consistent way.

Computes contribution margins for a line of products from revenues, cost of goods, and operating expenses. Expenses are segmented by department (marketing, product development, customer support) and by cost factors (manpower costs, program costs and overhead costs).

Computes marketing program contribution margins by allocating revenue to marketing programs that touched a customer within a specified time before each order. Revenue allocation and contribution margin (also known as “return on marketing investment”) constitute a significant advance in marketing management methods in the past decade.

Interprets results of pricing tests to help set prices that optimize revenue or profitability. The simpler model analyzes a single product, a more complex version analyzes multiple interacting products that have cross-elasticities.
4.3.3 Operations Analysis

  Tracks flows of units and costs through business and manufacturing process flows, using transition rates between stages, yield rates, scrap rates, rework rates. Tracks units of input, work-in-process, and finished goods; total cost and cost per unit for input, processing, and output for each stage. A sales model and finished goods inventory targets drive a production model.

  Computes operating measures and profit margins for customer support for a line of products. Quantifies resources spent for support for each product and for each tracked support issue (such as installation, maintenance, applications assistance).

4.3.4 Other Management

  Ranks alternative courses of action based on multiple evaluation criteria. You can specify the alternative courses of action, the evaluation criteria, the relative weights of the criteria, and the score for each alternative on each criterion. The Advanced version optionally includes a risk analysis of the alternatives, and provides a version of scores that takes into account the uncertainty in quality of outcomes and your risk tolerance / risk aversion.

  Tracks and scores candidates for an open employment position using your evaluation criteria, criterion weights, and ratings for each candidate. It is particularly useful as a compact way to organize assessments of the candidates.

The most unique aspect of these models is the ease with which modelers can author, understand, audit, and customize them.

5 Conclusion

The conventional spreadsheet paradigm, though enhanced in many important ways, has remained essentially the same since the late 1970s. Survival of the design concepts for three decades, with as little basic change as spreadsheets have seen, is a tribute to the strength of the spreadsheet concept. The spreadsheet paradigm – especially enhanced by pivot tables and graphs – remains excellent for a wide range of reporting tasks. However, conventional spreadsheet authoring environments lag behind the requirements imposed by larger and more complex applications for which spreadsheets are now used.

ModelSheet Authoring defines a new paradigm for authoring quantitative models. The root change is that ModelSheet combines programming and sheet layout in a more productive way in the authoring process. Formulas are expressed in terms of named variables, not cell addresses. Formulas in analysis variables are attached to regions that are defined by dimensions and time series. ModelSheet automates roll-up and splitting operations using information about the “types” of variables. Separating authoring of model logic from authoring sheet layout frees each authoring process to be unconstrained by the other. ModelSheet makes no changes in report delivery; instead it lays out web workbooks and translates them into final reports using the polished and familiar reporting medium of conventional spreadsheets.
Named variables, symbolic formulas, time series and dimensions have appeared in other products that are cousins of spreadsheets. Just as new recipes use old ingredients in a different way, ModelSheet Authoring combines old ideas and some new ones to make an advance in spreadsheet modeling.