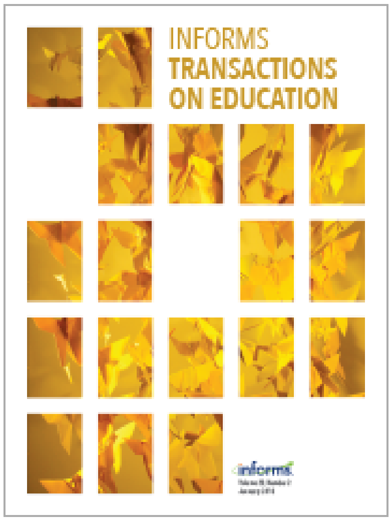

INFORMS Transactions on Education

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

An Electronic Teaching Assistant for Basic Operations Management Methods

Ralph J. Scheubrein, Sadan Kulturel-Konak,

An Electronic Teaching Assistant for Basic Operations Management Methods

Ralph J. Scheubrein
Operations and Logistics Management
Freiberg University
te@scheubrein.com

Sadan Kulturel-Konak
Management Information Systems
Penn State Berks
sadan@psu.edu

Abstract

Software packages are frequently used to support the teaching of quantitative methods. In one possible approach, the educator demonstrates how to use a software package which implements the method considered in the lecture. In an alternative approach, students have to implement the considered method on their own. A popular type of software package used in Master of Business Administration classes for the second approach is the spreadsheet, since students can "program" in an intuitive way by entering formulas in the cells of the worksheet. While letting students implement a method on their own normally takes much longer than just a demonstration by the educator, it can be argued that the students develop a better understanding of the method through active learning. In this context, the spreadsheet-based tutoring system ExcelTutor was developed to help students learn the steps necessary to compute the results of a specific method. After a student has formulated the method on his or her own, ExcelTutor verifies the correctness of the final results and provides feedback on the time needed for the implementation.

1. Introduction

When teaching quantitative methods of operations management in Master of Business Administration (MBA) classes, there are various educational objectives that instructors would like to achieve, including the following (see Anderson, 2001, for a comprehensive treatment of educational objectives in general):

1. Students should be aware of methods to solve specific business problems.
2. Students should be able to implement a method in a software package.

In the first case, a method can be treated as a black box. Students have to verify that the prerequisites for applying the method are met and must be able to interpret the results. For common methods, there is often a software implementation available, which the instructor can use in the classroom for demonstration. If the software is especially designed for educational purposes, typically not only it computes the result, but also tries to give some insight into the inner workings of the method. In the field of operations management, such educational software accompanies the book of Stevenson (2005). For this book, implementations of various methods are available both online and on the companion CD-ROM (see the publisher's web page[1] under the menu entry "Interactive OM" resp. "Excel Templates"). For example, to explain the Exponential Smoothing method, the Microsoft Excel template "Chap3.xlt" can be used. In this template, a chart is used to visualize the effects of changing the smoothing parameter *a*: the closer the value of the parameter is to 0, the greater the smoothing and the less the responsiveness. A student can also use this worksheet as a black box (i.e., a given data set can be entered, and Microsoft Excel computes the result of exponential smoothing).

[1] http://highered.mcgraw-hill.com/sites/0072869054/student_view0/

The second educational objective is naturally more demanding for the educator and students. First, the educator has to explain more details of the method, and students have to understand them. Second, the educator and students ought to have knowledge of a software package which provides the basis of the implementation. Applying a specialized software package to each problem type (e.g., one for optimization, one for simulation, and one for data mining) would provide students with the latest technology available, but learning how to work with several software packages would also be time consuming.

In particular in MBA classes, using spreadsheets - namely Microsoft Excel - seems to be a good compromise, considering the time necessary to learn how to handle this software package and the range of problems which can be addressed in the classroom. In addition, many real life business problems can be analyzed with spreadsheet-based systems. To be able to implement such analytical systems, a very first step could be that the student learns how to implement a single method. For this early learning stage, the system ExcelTutor was created to demonstrate to students the computational steps of a specific basic method. If a topic like forecasting is discussed in the classroom, some ExcelTutor workbooks implementing forecasting methods can be given to the students as a home or lab assignment. This means that the classroom time is saved for discussions while students can acquire the technical skills by working on that particular assignment.

In Section 2, it is demonstrated how a student learns with ExcelTutor workbooks. In Section 3, it is shown how an educator creates such workbooks. At the end of this article, a short conclusion is given.

2. Student Mode

In this section, first a simple method is presented to illustrate the functionality of ExcelTutor. The second example uses some lesser-known functions of Microsoft Excel to show that more complicated methods can also be taught with ExcelTutor.

2.1. Exponential Smoothing

To accompany this description, it might be convenient to download the zip-archive(2) of this example and run the file "ExponentialSmoothing.xls". ExcelTutor is implemented in "Visual Basic for Applications" (VBA) and you should allow Microsoft Excel to execute the macros of this workbook. When opening the file, if you are not automatically re-directed to the "Welcome Student!" worksheet with general information about ExcelTutor, you must first change the security level of Microsoft Excel to "Medium". This can be configured in the "Tools" menu, "Macro" entry, "Security..." sub entry, "Security Level" tab.

First, the system displays a general welcome message and the user has to click the "Start extending your knowledge!" button. ExcelTutor switches to the "Exercise" worksheet with the problem highlighted. To start a new exercise, click the "(Re-)Start" button in the upper left corner, and the screen will appear as shown in Figure 1. In an ExcelTutor worksheet, the input of the method is in green cells, while the output is expected in one or more yellow cells (for a discussion on this color-coding scheme, see Scheubrein,2003). If the computation is very simple, the student can just enter the appropriate formula into the output cell. More typically, a student would break up the computation into intermediate steps in a separate area of the worksheet. ExcelTutor assumes that this area is "below" the problem definition (i.e., below row 17 in the example shown in Figure 1).

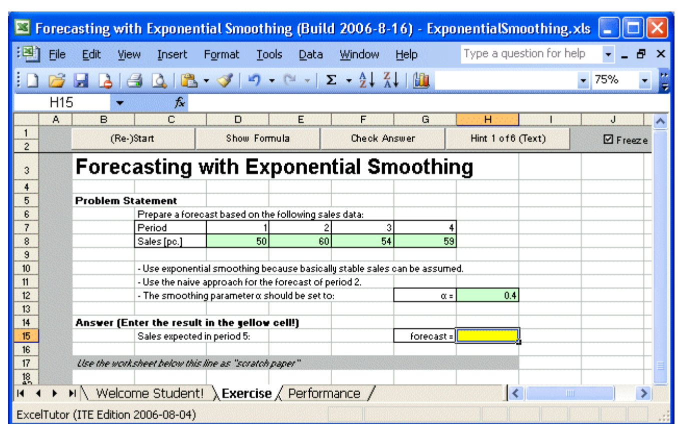

Figure 1: Initial worksheet with problem statement and question

(2) http://ite.pubs.informs.org/Vol7No1/ScheubreinKulturelKonak/ExcelTutor.zip

The following recursive formula is generally used to compute exponentially smoothed values based on a time series with the values *Ai* (e.g., Stevenson, 2005, p. 75):

$$F_t = (1-\alpha)F_{t-1} + \alpha A_{t-1}$$

where

F_t	=	Forecast for period t
F_{t-1}	=	Forecast for period $t-1$
$\alpha \in [0;1]$	=	Smoothing parameter
A_{t-1}	=	Actual observed value for period $t-1$

To complete this formula, it is necessary to define the start of the recursion (i.e., how to compute the first forecast). In this example, the so-called naive approach is used, which simply sets *F2* equal to *A1*. The steps to apply this formula to the problem are straightforward and are demonstrated by ExcelTutor when clicking on the "Hint" button in the upper right corner. After clicking on the "Hint" button for the first time, the user is prompted to enter the formula to compute "*1-a*" in cell H28 (see Figure 2). Clicking on the same button a second time reveals the formula needed (see Figure 3).

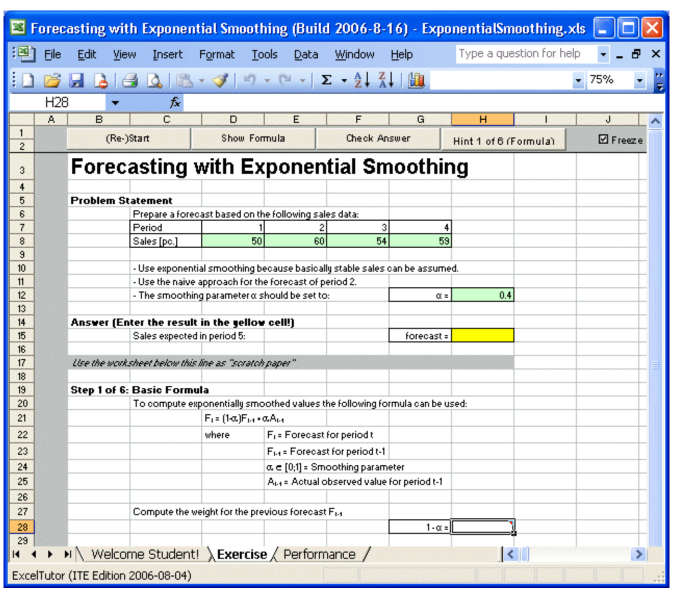

Figure 2: What to do in the first step

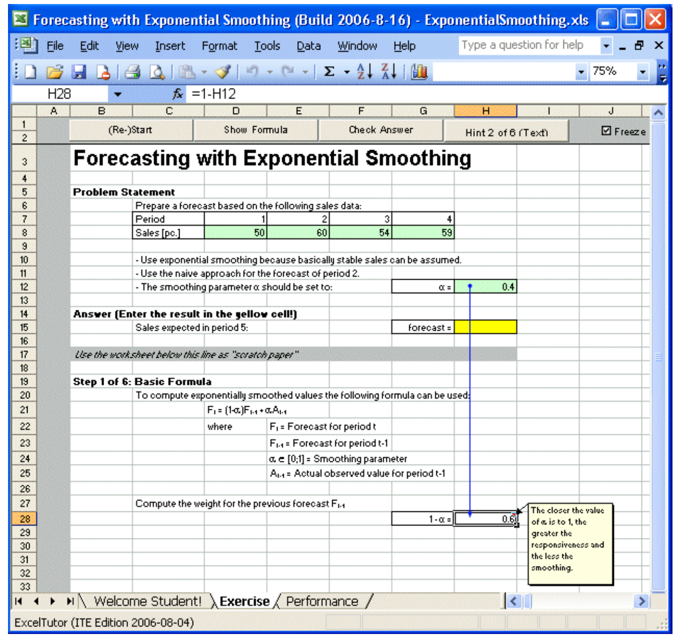

Figure 3: Formula of the first step

The student can click on the "Hint" button several times to learn all the steps and formulae necessary to solve the problem (see Figure 4 for an animation).

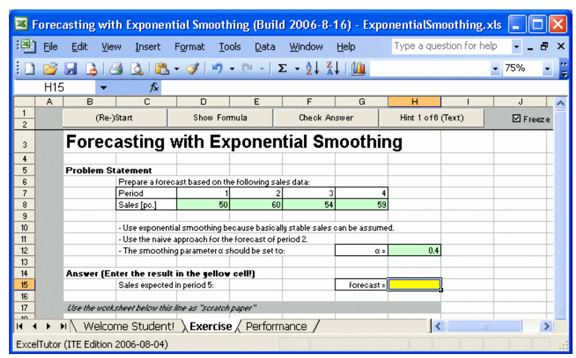

Figure 4: Animation showing all the steps necessary to compute the ultimate result

In addition to computing the result, there is also a suggestion for the spatial layout of a worksheet for exponential smoothing (see Figure 5). While ExcelTutor focuses on teaching the computational steps of a method, this suggestion might be useful for students who are interested in integrating such basic methods into larger spreadsheet-based decision support systems. These students should be aware of how to take advantage of the functionality of a spreadsheet. In this example, the formula implementing the recursion (cell F52) with an absolute reference to the smoothing parameter (cell E49) can simply be dragged to the right to fill the table.

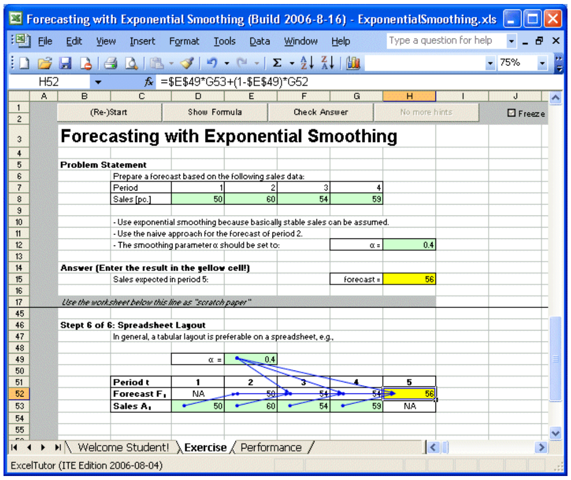

Figure 5: Suggestion for the spatial layout of a worksheet for exponential smoothing

After the student has inspected and understood the formulae provided in the hints, he or she should finally be able to re-start and input the correct formulae without any hints. With respect to the final exam, it is necessary to implement the method correctly and as fast as possible. To verify the correctness of the result (i.e., one or more "yellow" cells), the "Check Answer" button is used. The time that was needed to get the correct answer is automatically recorded on the "Performance" worksheet (see Figure 6). If the student is not satisfied with his/her performance it is possible to either go immediately back to the "Exercise" worksheet to try again or to save the workbook with the performance table to try again later ("Save Workbook" button). In addition, the "Clear Log" button is provided to erase the performance table.

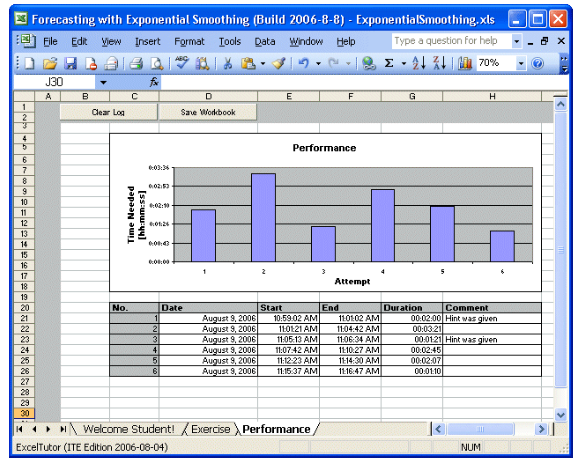

Figure 6: The time necessary to find the correct result is recorded as feedback for the student

2.2. Service Levels

While the method considered in the previous example was very simple, this example shows a more advanced application. This problem deals with inventory control, in particular, the system's ability to meet the demand. Two frequently used performance measures (e.g., Nahmias, 2001, p. 265) are:

- Type 1 Service: α represents the probability of having enough items in stock to meet the demand.
- Type 2 Service: β represents the proportion of the demand that is met from stock.

A basic approach is modeling the demand as a continuous random variable, which is normally distributed with mean μ and standard deviation σ. Let Q denote the number of items in stock of a certain product type and Q^* the minimum quantity in stock to meet a given service level for that product type. In the following, let y be a probability value, x be a given demand for that product type, and $z=(x-\mu)/\sigma$ be its corresponding standardized variate. Table 1 summarizes some functions useful in this context.

Table 1: Selected functions and their equivalents in Excel

Mean m and Standard Deviation s	Name	Function	Function in Excel	Inverse Function in Excel
non-standardized	normal density function	$f_{m,s}(x)$	NORMDIST(x,m,s,FALSE)	NA
	normal distribution function	$F_{m,s}(x)$	NORMDIST(x,m,s,TRUE)	NORMINV(y,m,s)
standardized, i.e. m = 0 and s = 1	normal density function	$f_{0,1}(z)$	NORMDIST(z,0,1,FALSE)	NA
	normal distribution function	$F_{0,1}(z)$	NORMSDIST(z) or NORMDIST(z,0,1,TRUE)	NORMSINV(y) or NORMINV (y,0,1)
	loss function	$L_{0,1}(z)$	use functions for normal density and normal distribution (see below)	use a lookup table or goal seeking

To compute the necessary stock manually, typically a lookup table showing z, $F0,1(z)$, and $L0,1(z)$ is used (e.g., Nahmias, 2001, Table A-4 in the appendix). Using a two-step approach, such a table can be applied to problems with a mean not equal to 0 or a standard deviation not equal to 1. For a type 1 service, it is sufficient to look at the probability that the demand is not higher than the number of items in stock Q, i.e., $F_{\mu,\sigma}(Q)$. To meet the specified service level a, $F_{\mu,\sigma}(Q)>=\alpha$ has to be true and the minimum stock can be computed by $Q^*=F^{-1}_{\mu,\sigma}(\alpha)$. If a lookup table for z and $F0,1(z)$ is available, the necessary stock Q^* can be computed as follows:

$$z = F_{0,1}^{-1}(\alpha)$$
$$Q^* = \mu + z \cdot \sigma$$

For a type 2 service, the standardized loss function can be used (e.g., Nahmias, 2001, p. 270):

$$L_{0,1}(z) = \int_z^\infty (t-z) f_{0,1}(t)\,dt = f_{0,1}(z) - z\left(1 - F_{0,1}(z)\right)$$

If a lookup table for $L0,1$ (or of $f0,1$ and $F0,1$) is available, the necessary stock Q^* can be computed as follows:

$$z = L_{0,1}^{-1}\left(\frac{\mu(1-\beta)}{\sigma}\right)$$
$$Q^* = \mu + z \cdot \sigma$$

Figure 7 shows the corresponding ExcelTutor module "ServiceLevels.xls" in the zip-archive(3). For a type 1 service level, the function NORMINV can be used directly. For a type 2 service level, NORMDIST and goal seeking is combined to compute $L^{-1}_{0,1}$ (for a discussion of the goal seeking feature in Microsoft Excel see Troxell, 2002).

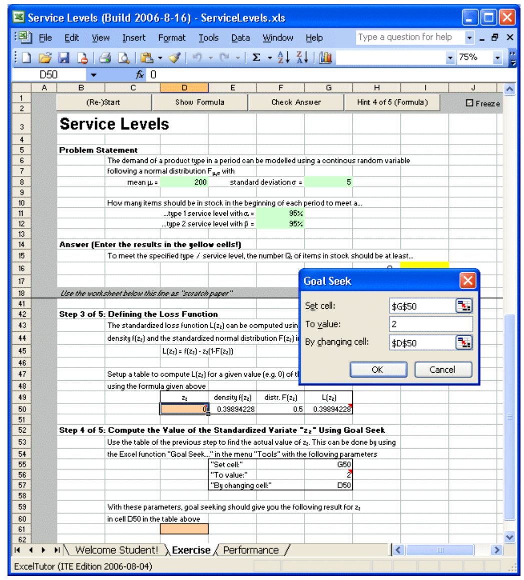

Figure 7: Computation of the stock necessary to meet specified service levels

3. Educator Mode

Students normally get a "sealed" workbook, i.e., a workbook in which the problem statement and the solution are stored on a hidden worksheet. For authoring a new workbook, this compressed archive(4) holds a second version of ExcelTutor with the exponential smoothing problem explained in Section 2.1. When starting the file "ExcelTutorForAuthors.xls" the screen in Figure 8 is displayed. This screen provides instructions for authoring new problems.

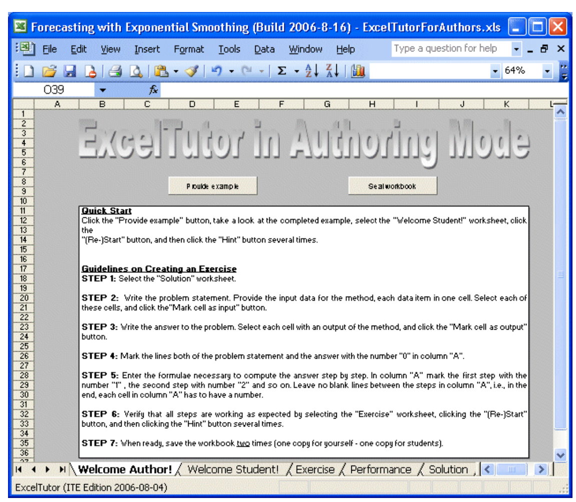

Figure 8: Welcome screen for an author

(3) http://ite.pubs.informs.org/Vol7No1/ScheubreinKulturelKonak/ExcelTutor.zip
(4) http://ite.pubs.informs.org/Vol7No1/ScheubreinKulturelKonak/ExcelTutorForAuthors.zip

New problems are authored on the "Solution" worksheet (see Figure 9). This worksheet basically looks like the worksheet presented to a student. The first difference is that in the first column "A", the author has to define which step each row belongs to. The problem statement and the question have to have the number 0 in that column. The second difference is that there are other buttons in the header. ExcelTutor needs to know which cells are input and which are output. To mark a cell accordingly, select the cell and click the "Mark cell as input" or "Mark cell as output" button. Finally, to start a new problem, the "Delete" button is provided to erase the worksheet completely.

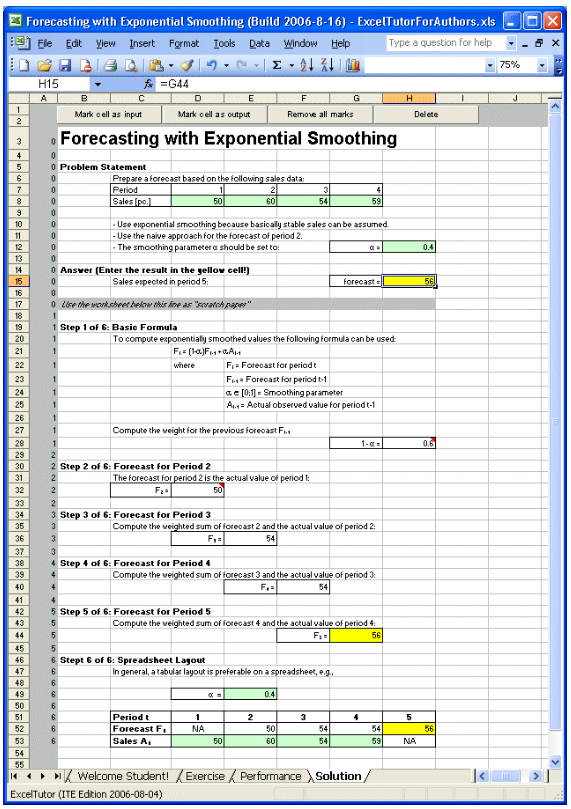

Figure 9: Definition of the steps in authoring mode

After having authored a new problem, the author can test it by selecting the "Exercise" worksheet, clicking the "(Re-)Start" button, and then clicking the "Hint" button several times. Finally, the author should store two copies of the new workbook. The one that will be given to students should be sealed by clicking on the "Seal workbook" button on the "Welcome Author!" worksheet (see Figure 8). The other one should be stored for future improvements. If this unsealed copy should get lost, a completed solution may simply be copied from the "Exercise" worksheet of a sealed copy to the "Solution" worksheet of an unsealed copy and the lines numbered accordingly.

4. Conclusion

ExcelTutor can be used to teach students basic methods of operations management. The creation of a workbook to demonstrate a method is simple and could, for example, be done by a teaching assistant. This approach seems reasonable if the ultimate result can be computed in a few "steps" (e.g., about seven steps - see classically Miller, 1956). Nevertheless, when using non-trivial functions of Excel (e.g., statistical functions as shown in section 2.2) many methods taught in a basic operations management course can be implemented within this limit.

ExcelTutor is intended to be used as an element of a broader educational approach. The main objective of this approach is to enable MBA students, i.e. future end-users and knowledge workers without explicit training in software engineering, to build spreadsheet-based analytical tools on their own. Such a system is typically used by its creator as a personal productivity tool. Major steps in this educational approach spanning several classes include:

1. In the very beginning, the students will work on the "Excel Tutorial" of Albright (2001a). This tutorial is a Word document with many embedded spreadsheets. Completing the exercises in this document ensures that every student has a basic knowledge of functions (e.g., SUMPRODUCT) and shortcuts (e.g., for navigation) available in Excel.
2. Next, the students implement basic methods for operations management using ExcelTutor like the ones presented in this article. In parallel, the students should learn how to design a spreadsheet (see e.g. Raffensberger, 2003).
3. After having learned how these methods work, simple analytical tools can be implemented by packaging the basic methods in a user-friendly way (e.g., Ragsdale,2001, or Scheubrein, 2003). These systems are implemented using the front-end functions of Excel, i.e. formulae in cells.
4. The next step is to extend the basic functionality of Excel by using add-ins like the Solver (for an overview of add-ins see Grossman, 2002).

 © INFORMS ISSN: 1532-0545

5. Finally, more advanced decision support systems can be implemented by using the back-end of Excel with VBA. A major reference for this approach is Albright (2001b).

An MBA student who masters all these steps should be able to build personal productivity tools for various quantitative problems arising in his/her job. If such a spreadsheet-based system seems to be useful for the business for a significantly long time period and is used by several people, it can serve as a specification for a reimplementation by the information systems department. Such a reimplementation by professional software engineers would probably be done in a standard programming language like Java and could have features such as a web-interface and direct access to enterprise-wide databases. During the testing phase of such a large-scale system, the spreadsheet-based prototype can be used to evaluate test cases and compare the results of both systems.

Acknowledgements

The authors graciously acknowledge to the guest editor of this special issue of ITE and the two referees who provided invaluable comments and suggestions.

References

(all web links verified on August 8, 2006)

Albright, C. (2001a), "Excel Tutorial to Improve Your Efficiency", http://www.kelley.iu.edu/albrightbooks/#Tutorial

Albright, C. (2001b), "VBA for Modelers - Developing Decision Support Systems Using Microsoft Excel", Duxbury, Pacific Grove, CA http://www.kelley.iu.edu/albrightbooks/VBA_Home.htm

Anderson, L. W. et al. (eds., 2001): "A Taxonomy for Learning, Teaching, and Assessing - A Revision of Bloom's Taxonomy of Educational Objectives", Longman, New York.

Grossman, T. A. (2002), "Spreadsheet Add-Ins for OR/MS", *ORMS Today*, http://www.lionhrtpub.com/orms/orms-8-02/frsurvey.html

Miller, G. A. (1956), "The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information", *The Psychological Review*, Vol. 63, pp. 81-97, http://www.well.com/~smalin/miller.html

Nahmias, S. (2001), "Production and Operations Analysis", 4th ed., McGraw-Hill/Irwin, Boston, MA.

Raffensperger, J. F. (2003), "The New Guidelines for Writing Spreadsheets", http://www.mang.canterbury.ac.nz/people/jfraffen/spreadsheets/index.html

Ragsdale, C. T. (2001), "Teaching Management Science With Spreadsheets: From Decision Models to Decision Support", *INFORMS Transactions on Education*, Vol. 1, No. 2, http://ite.pubs.informs.org/Vol1No2/Ragsdale/Ragsdale.php

Scheubrein, R. (2003), "Elements of End-User Software Engineering", *INFORMS Transactions on Education*, Vol. 4, No. 1, http://ite.pubs.informs.org/Vol4No1/Scheubrein/index.php

Stevenson, W. J. (2005), "Operations Management", 8th ed., McGraw-Hill/Irwin, Boston, MA, ("Tour the DVD" http://highered.mcgraw-hill.com/sites/0072869054/information_center_view0/tour_the_dvd.html , "Excel Templates" http://highered.mcgraw-hill.com/sites/0072869054/student_view0/excel_templates.html , and "Interactive OM" http://highered.mcgraw-hill.com/sites/0072869054/student_view0/interactive_om.html)

Troxell, D. S. (2002), "Optimization Software Pitfalls: Raising Awareness in the Classroom", *INFORMS Transactions on Education*, Vol. 2, No. 2, http://ite.pubs.informs.org/Vol2No2/Troxell/