
Development of an e-textbook using \LaTeX and PSTricks

David M. Tulett

Abstract

For a course on decision modeling (linear, integer, and goal programming, networks, and decision trees) I created an e-textbook using \LaTeX and PSTricks. I will discuss why I chose these programs, how I used them, and why I decided to make the final product “open access”. The full PDF can be downloaded from <http://stor.mun.ca/handle/123456789/37463>.

1 Introduction

A wide-sweeping curriculum change in the undergraduate business programs at Memorial University necessitated the development of many new courses, including Business 2400, Decision Modeling. The introduction of the new courses was staggered, beginning in 2010, with Business 2400 first being offered in Fall Term, 2011. Here is the course description from the 2011–2012 university calendar:

2400 Decision Modeling provides an introduction to: spreadsheet modeling; linear optimization and the related topics of integer, assignment, and transportation models; and decision analysis including payoff matrices, decision trees, and Bayesian revision. All topics will be taught within the context of business applications. [4]

This new course replaced the former Business 4401 and it differed in the following ways:

1. Business 2400 comes one year earlier in the curriculum than Business 4401 did. Since 4401 (and 4500 Finance) acted to “weed out” students from the program, it was felt that 2400 should come earlier than 4401 did.
2. Business 2400 continued the de-emphasis of the learning of algorithms that began when Business 4401 was created 15 years earlier. The only algorithms taught in Business 2400 are: solving two-variable optimization problems graphically; the graphical method for solving the minimal spanning tree problem; and the rollback procedure for solving decision trees.
3. As the name implies, there is now an increased emphasis on the modeling of problems. In most cases this means defining a set of variables, and then stating an objective function to be maximized or minimized subject to a set of constraints.

The author is grateful for funding from a Teaching Fellowship to attend the 2016 TUG conference.

4. For the solution of formulated models, only Excel is used. All students at Memorial receive a copy of Microsoft Office, which includes Excel, hence there is no new software to download. Also, learning Excel has to be done anyway, so the only new things would be some specific mathematical functions and the use of the Solver.

When the course was first offered in the Fall Term of 2011, we adopted a textbook for it: *Managerial Decision Modeling with Spreadsheets* by Render *et al.* [5] At the time, the book sold in the campus bookstore for about \$156. (All monetary figures in this article are in Canadian currency.) Alternatively, it could be obtained as an e-book for about \$70, but this comes with only a six-month licence.

Even though we had adopted a textbook, before the course had even begun I had started to write a document with the course title *Decision Modeling*. Principally, this was because the course was about to be offered in Winter 2012, by what was then known as “distance education” (as of 2016, this is now called *online learning*). For the students in this course, my document would substitute for the lectures that they would miss by not being on-campus. Also, I felt that I could improve upon the textbook’s coverage of some topics, such as decision analysis (the use of decision trees). Five years later, the *Decision Modeling* document has become an open access stand-alone e-textbook, and the rest of this article describes how this happened.

In section 2, I describe the beginning of the writing: why \LaTeX was chosen, merging of material from earlier courses, and development of new material. In section 3, I describe what I found useful when I was presented with options for completing various tasks in writing the document. Finally in section 4, I describe why this document became open access.

2 Initial development

I have been using \LaTeX since the early 1990s, so I am very familiar with it and its graphical cousin, PSTricks. I am also very familiar with Microsoft Word, it being the standard for word processing where I work. Indeed, I use Word anytime that I need to collaborate with a co-worker, be it for administrative purposes or for writing a journal article. Perhaps if the writing of *Decision Modeling* had been a collaborative effort by multiple professors the document would have had to be written using Word. However, I was doing it on my own, so the choice was mine.

From previous courses I had amassed a large amount of material, mostly written in \LaTeX , which could, with modification, be embedded within the

new document, partially fulfilling its requirements. For this reason alone it would have made sense to create the *Decision Modeling* document in L^AT_EX, but there are two other main reasons.

1. L^AT_EX looks better than Word. This is especially true when creating mathematical expressions.
2. Because things like section numbers are never entered by the user in L^AT_EX, it becomes much easier to move things about, letting L^AT_EX figure out how everything (sections, figures, tables, footnotes) is to be renumbered.

All this being said, there is one important development in L^AT_EX that was needed to make the finished product useful for distribution as an e-document, and that is the creation of *pdfL^AT_EX*. Back in the 1990s, a .tex file was compiled to create a .dvi file. For anyone with a T_EX system, all one had to do was print the .dvi file. However, members of the general public could not be expected to have this ability, and even when stand-alone free .dvi readers came out, it was still a barrier to have to expect people not interested in L^AT_EX to download the reader. Once the ability to easily compile a .tex file to a PDF became available, it was a major step forward. While Microsoft Office is a standard piece of software where I work, once a document has been converted to PDF its original source becomes irrelevant, be it Word, WordPerfect, Libre Office Writer, or L^AT_EX.

Hence, for all these reasons, it made sense to create this document using L^AT_EX for conversion to PDF. As I have mentioned, Excel is used as the prominent method of solution. For any parts of the document which used Excel, nearly everything had to be written from scratch. In other places I was able to import previously created material, but even here substantial alterations had to be made. The alterations were of two main types:

1. In cutting-and-pasting from several source documents, I was greatly aided by L^AT_EX's logical design, which frees the user from worrying about the numbering of sections, figures, tables, and footnotes. However, the text had to be extensively edited for phrases such as "as we saw in the previous chapter", which would probably now be an anachronism. Also, there needed to be consistency about notation, avoiding, for example, x_1 in one place while using X_1 in another. More subtle than these things, though, would be the existence of ideas which presume a knowledge which might not apply to readers of the new document.
2. Some of the older documents were based on earlier technology, such as the use of L^AT_EX's

native picture environment (rather than the far more sophisticated PSTricks). Over time, these things have been updated.

At the outset of the Fall of 2011, the *Decision Modeling* document consisted of the following chapters, with much content to be completed:

1. **Introduction** Back in 2011, this was a very short chapter, consisting of a look at the paradigm of problem identification, modeling, solution, and implementation; a review of some key Excel functions; and a brief look at professional associations.
2. **Elementary Modeling** This chapter essentially replicated one used in Business 4401, so this required little work.
3. **Applications of Linear Models** Many of the examples in this chapter had been used in previous courses, but they had never been solved in Excel.
4. **Sensitivity Analysis** Most of the concepts had been seen before, but the pictures to illustrate these concepts had been created in the native L^AT_EX picture environment rather than PSTricks. Also, all computer output needed to be done in Excel.
5. **Network Models** These models are for the assignment, transportation, transshipment, minimal spanning tree, maximal flow, and shortest path problems. The minimal spanning tree problem has a very easy graphically-based algorithm for its solution. For the other five problems, this course stresses the use of Excel, in contrast to previous courses in which these problems had purpose-built algorithms. Because of this, this chapter needed to be written almost from scratch. Nothing had been written on this chapter as of September 2011.
6. **Integer Models** Much of the theory had been described in material written for previous courses, but none of the models had been solved in Excel.
7. **Goal Programming and Nonlinear Models** Same problem as the previous chapter: theory, but no Excel.
8. **Decision Analysis I** Mostly done already, but some Excel work required.
9. **Decision Analysis II** Mostly done already, but some Excel work required.

In addition to the work that needed to be done to complete all these chapters, another requirement was to create end-of-chapter problems for student completion, and to make the solutions for these problems.

3 Making the document

The Fall Term proceeded with the Render *et al.* textbook, on which all course assignments were based. As a supplement, students in my sections could download a PDF then called *Business 2400 Decision Modeling Course Manual* (a document title spread over three lines) which had no Chapter 5, and with the problems noted above in the other eight chapters. An immediate first priority was to have something written for Chapter 5. One of these models is the Assignment Problem, for which a small example is presented here, along with its mathematical model and its spreadsheet model.

3.1 Assignment Problem example: Assigning 3 jobs to 3 machines

Suppose that we have three jobs, and three machines on which these jobs will be done. Each machine will do just one of the three jobs. All three machines are capable of doing each job, but with differences in performance. We can think of these differences in terms of cost (which could be time rather than dollars). Suppose that the costs to assign each job (row) to each machine (column) are as follows:

| | Machine | | |
|-------|---------|----|----|
| | 1 | 2 | 3 |
| Job 1 | 30 | 20 | 18 |
| Job 2 | 17 | 40 | 21 |
| Job 3 | 25 | 32 | 28 |

By inspection, the minimal cost solution is to assign job 1 to machine 2, assign job 2 to machine 1, and assign job 3 to machine 3, for a total cost of $20 + 17 + 28 = 65$. To solve this as a mathematical model, we define the meaning of $X_{i,j}$ for all pairs (i, j) :

$$X_{i,j} = \begin{cases} 1 & \text{if job } i \text{ is assigned to machine } j \\ 0 & \text{otherwise} \end{cases}$$

$$i = 1, 2, 3 \quad j = 1, 2, 3$$

The reason for using the numbers 1 and 0 becomes clear when we write the model. For example, if job 2 is assigned to machine 3 (i.e. $X_{2,3} = 1$), then the cost is $21(1) = 21$. If job 2 is *not* assigned to machine 3 (i.e. $X_{2,3} = 0$), then the cost is $21(0) = 0$. Hence, whether or not job 2 is assigned to machine 3, we incur a cost of $21X_{2,3}$.

Hence the objective function is:

$$\begin{aligned} \text{minimize} \quad & 30X_{1,1} + 20X_{1,2} + 18X_{1,3} + \\ & 17X_{2,1} + 40X_{2,2} + 21X_{2,3} + \\ & 25X_{3,1} + 32X_{3,2} + 28X_{3,3} \end{aligned}$$

Every job must be assigned to a machine, hence for each job i one of $X_{i,j}$'s will be 1 (and the other two

will be 0), hence the sum will be 1:

$$\begin{aligned} X_{1,1} + X_{1,2} + X_{1,3} &= 1 \\ X_{2,1} + X_{2,2} + X_{2,3} &= 1 \\ X_{3,1} + X_{3,2} + X_{3,3} &= 1 \end{aligned}$$

Every machine must have a job assigned to it, hence for each machine j one of $X_{i,j}$'s will be 1 (and the other two will be 0), hence the sum will be 1:

$$\begin{aligned} X_{1,1} + X_{2,1} + X_{3,1} &= 1 \\ X_{1,2} + X_{2,2} + X_{3,2} &= 1 \\ X_{1,3} + X_{2,3} + X_{3,3} &= 1 \end{aligned}$$

Finally, each variable must be 0 or 1.

$$\text{all } X_{i,j} \in \{0, 1\}.$$

In one sense this is a specialized type of linear programming problem, but it seems to violate one of the assumptions of linear programming — that all variables be continuous, rather than integer. However, it turns out that the assignment problem is naturally integer. By this, we mean that the solution will only contain 0/1 variables, even when these have not been specifically required. Hence, any software for general linear programming will solve an assignment problem.

We will use the Solver in Excel to solve this type of problem. Indeed, the rectangular array paradigm of Excel is very useful for this type of problem, where the cost data is in this format in the first place.

Since the cost data are in a 3 by 3 array, we can also use a 3 by 3 array for the values of the variables. Note that the SUMPRODUCT function is happy with this; here it's an array times an array on a cell-by-cell basis, not the dot product of one row with another row. In this example it's B3 times B10 plus C3 times C10 and so on up to D3 times D12. This type of product is not the same as matrix multiplication. Here is the setup in formula mode on the spreadsheet, before entering the Solver (the = signs in row 7 and column F are created by typing '=').

| | A | B | C | D | E | F | G |
|----|----------------------------|-------------|-------------|-------------|-------------|---|---|
| 1 | Assignment | | Machine | | | | |
| 2 | Problem | 1 | 2 | 3 | sum | | |
| 3 | Job 1 | | | | =SUM(B3:D3) | = | 1 |
| 4 | Job 2 | | | | =SUM(B4:D4) | = | 1 |
| 5 | Job 3 | | | | =SUM(B5:D5) | = | 1 |
| 6 | sum | =SUM(B3:B5) | =SUM(C3:C5) | =SUM(D3:D5) | | | |
| 7 | | = | = | = | | | |
| 8 | | 1 | 1 | 1 | | | |
| 9 | | | | | | | |
| 10 | Total Cost | 30 | 20 | 18 | | | |
| 11 | =SUMPRODUCT(B3:D5,B10:D12) | 17 | 40 | 21 | | | |
| 12 | | 25 | 32 | 28 | | | |

In the Solver we ask it to minimize A11 by changing variable cells B3:D5, subject to the three constraints B6:D6 = B8:D8, and the three constraints E3:E5 = G3:G5. We click on the “Make unconstrained variables non-negative” box, and ask for the problem to be solved using the “Simplex LP”. Solving the model we obtain:

| | A | B | C | D | E | F | G | |
|----|------------|----|---------|----|-----|---|---|--|
| 1 | Assignment | | Machine | | | | | |
| 2 | Problem | 1 | 2 | 3 | sum | | | |
| 3 | Job 1 | 0 | 1 | 0 | 1 | = | 1 | |
| 4 | Job 2 | 1 | 0 | 0 | 1 | = | 1 | |
| 5 | Job 3 | 0 | 0 | 1 | 1 | = | 1 | |
| 6 | sum | 1 | 1 | 1 | | | | |
| 7 | | = | = | = | | | | |
| 8 | | 1 | 1 | 1 | | | | |
| 9 | | | | | | | | |
| 10 | Total Cost | 30 | 20 | 18 | | | | |
| 11 | | 65 | 17 | 40 | 21 | | | |
| 12 | | 25 | 32 | 28 | | | | |

As we saw earlier, we see from the Solver output that the minimal cost solution is to assign job 1 to machine 2, assign job 2 to machine 1, and job 3 to machine 3, with a total cost of 65.

3.2 Typesetting the above in L^AT_EX

The above description uses the `tabular` environment for the cost data, and the `array` environment for the objective function and equations. When it comes to doing the Excel output, it would have been possible to use the `tabular` environment to mimic a spreadsheet, including the use of colour, but this would have been slow and cumbersome. Instead, I made the file in Excel, highlighted the range that I wanted printed, went to File/Print, set the Printer to Adobe pdf, made the Settings “Print Selection” and “Fit All Columns on One Page”, and under Page Setup, then Sheet, then Print, clicked the boxes for “Gridlines” and “Row and column headings”. I named this file `Assignment2.pdf`, then using Adobe Acrobat, this file was cropped and saved. It was imported into the book’s `.tex` file as:

```
\begin{center}
\includegraphics[scale=0.9]{Assignment2.pdf}
\end{center}
```

This file needed to be re-scaled to 90% of its original size in order not to spill too much into the margin. This re-scaling is a visual trial-and-error process. It wasn’t needed for the final numerical workbook image:

```
\begin{center}
\includegraphics{Assignment3.pdf}
\end{center}
```

When importing Excel files in this manner, the ribbon with its words (File, Home, Insert, etc.) and all the icons does not appear. Normally, I think that this is to be preferred — it emphasizes that the most important part of a workbook is what lies in the rows and the columns. If including the ribbon is desired, we can use the Print Screen command and then import this file into Adobe Acrobat. In my

document, I do this only once at the outset, to show what the ribbon looks like.

With this section on the assignment problem, plus other sections for the other network models, I finally had a first draft of Chapter 5 completed before the outset of the Winter Term, 2012. Next came the inclusion of dozens of PDFs created by cropping files created in Excel for the many problems covered throughout the document. A major piece of work remained until 2015, the creation of the graphical images needed for Chapter 4 (Sensitivity Analysis). The earlier work on which this chapter was based included graphics made in the native L^AT_EX `picture` environment. While this environment was better than nothing, it was very inadequate. Everything was in black-and-white, with very limited ability to draw curved lines. Even straight lines were limited to horizontal and vertical lines, and lines whose rise over the run could be expressed as $a : b$, where a and b are integers from 1 to 6 inclusive.

The way to improve the graphics is to use better software, such as PSTricks, which like L^AT_EX can be freely obtained from TUG. A good introduction is provided in Chapter 5 of *The L^AT_EX Graphics Companion* by Goossens *et al.* [3] A complete description of PSTricks is contained in *PSTricks: Graphics and PostScript for T_EX and L^AT_EX* by Voß. [8]

At the outset of Chapter 4, a problem is stated and solved. Here follows the problem description, model formulation, and graphical solution.

3.3 A two-variable example

3.3.1 Problem description

Wood Products Limited buys fine hardwoods from around the world from which they make specialized products for the quality furniture market. Two of their products are two types of spindles.

A type 1 spindle requires 6 cuts, then 4 minutes of polishing, followed by 6.5 minutes of varnishing. A type 2 spindle requires 15 cuts, then 4 minutes of polishing, followed by 4.75 minutes of painting. There is one cutting machine which can operate up to 135 cuts per hour. There is one polishing machine; allowing for maintenance it can operate up to 54 minutes per hour. Both the varnish and paint shops can only handle one spindle at a time. Because of a periodic need for high volume ventilation, the varnish and paint shops cannot be operated continuously. These shops are available for production 58.5 and 57 minutes per hour, respectively.

For each type 1 spindle produced, the company obtains a contribution to profit of \$3. For each type 2 spindle produced, the contribution to profit is \$4. How many spindles of each type should be

produced each hour so that the total contribution to profit is maximized?

3.3.2 Model

We define:

X_1 — number of type 1 spindles produced per hour
 X_2 — number of type 2 spindles produced per hour.

For reference, each constraint is identified by a word description on the left-hand side, and by a number in brackets on the right-hand side.

$$\begin{array}{llll}
 \text{maximize} & 3X_1 & + & 4X_2 \\
 \text{subject to} & & & \\
 \text{Cutting} & 6X_1 & + & 15X_2 \leq 135 \quad (1) \\
 \text{Polishing} & 4X_1 & + & 4X_2 \leq 54 \quad (2) \\
 \text{Varnishing} & 6.5X_1 & & \leq 58.5 \quad (3) \\
 \text{Painting} & & & 4.75X_2 \leq 57 \quad (4) \\
 & X_1 & , & X_2 \geq 0
 \end{array}$$

3.3.3 Graphical solution

Because of the two 4's in the polishing constraint, this constraint will be on a diagonal. Since it's \leq , the arrow indicating feasibility will point south-west. So, since $54/4 = 13.5$, having a 14 by 14 grid must contain the optimal solution. Using these boundaries, we obtain the values in Figure 1. The graph (displaying both numerical labels and the names of the constraints) is shown in Figure 2.

We see that constraints (1) and (2), i.e. the cutting and polishing constraints, are binding. The equations we need to solve are:

$$\begin{array}{rcl}
 6X_1 + 15X_2 & = & 135 \\
 4X_1 + 4X_2 & = & 54
 \end{array}$$

Multiplying the second equation by $6/4 = 1.5$ we obtain:

$$\begin{array}{rcl}
 6X_1 + 15X_2 & = & 135 \\
 6X_1 + 6X_2 & = & 81
 \end{array}$$

Subtracting the bottom from the top gives $9X_2 = 54$, and hence $X_2 = 6$. Therefore $4X_1 + 4(6) = 54$, hence $4X_1 = 30$, and therefore $X_1 = 7.5$. Putting $X_1 = 7.5$ and $X_2^* = 6.0$ into the objective function we obtain $OFV^* = 3(7.5) + 4(6) = 46.5$. The 7.5 type 1 spindles per hour simply means that we must produce 15 of them every two hours, hence the fractional solution is not of concern.

| | | | | | | |
|------------|----------|---|-----------|-------------|----------------|--------------|
| | | | | | First Point | Second Point |
| Cutting | $6X_1$ | + | $15X_2$ | ≤ 135 | (1) (0, 9) | (14, 3.4) |
| Polishing | $4X_1$ | + | $4X_2$ | ≤ 54 | (2) (0, 13.5) | (13.5, 0) |
| Varnishing | $6.5X_1$ | | | ≤ 58.5 | (3) $X_1 = 9$ | vertical |
| Painting | | | $4.75X_2$ | ≤ 57 | (4) $X_2 = 12$ | horizontal |

Figure 1: Spindle problem, numerical.

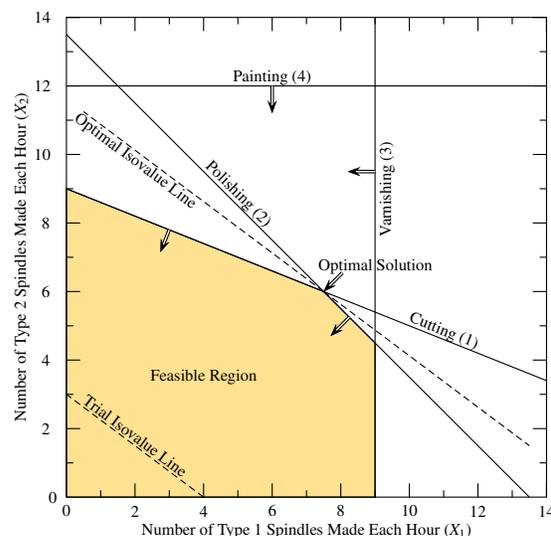


Figure 2: Spindle problem, graphical.

3.4 Doing the above in PSTricks

Everything in the preceding section is easy until we come to the graph. What we need is to be able to: draw straight lines at any angle; write text next to these lines at the same angle; draw arrows at right angles to these lines indicating which side is true for the inequality; fill in the polygon which represents the region in which all inequalities are true. PSTricks gives all these things, though a bit of geometry/trigonometry is needed to make it all work properly. Here are some of the special features of this picture, showing the relevant PSTricks code:

1. We have some lines which are neither horizontal nor vertical. If we take the Cutting constraint as an example, we have identified that the boundary of the inequality, which is the line $9X_1 + 15X_2 = 135$, passes through (0, 9) (on the vertical axis), and (14, 3.4) (on the right-hand side boundary). I found it useful to plot points as integers, so all algebraically determined points were multiplied by 100, making these coordinates (0,900) and (1400,340), but then to use a scaling command in PSTricks to make the graph for the page properly. For this graph, I

used a quarter scale, using the PSTricks command `\psset{unit=0.25pt}`, and then created the line using the PSTricks command

```
\psline(0,900)(1400,340) %(Cutting)
```

- The set of points for which all inequalities are true is called the *feasible region*. To identify this region, it is highlighted in colour with the built-in paint program in PSTricks. The problem is, we need to give PSTricks the set of coordinates which gives the vertices (“corners”) of the feasible region. This requires doing some successive solving of two equations in two unknowns to determine these points. They are the origin at (0,0); then where the vertical axis meets the boundary of the cutting constraint, which is (0,9); then where boundary of the cutting constraint meets the boundary of the polishing constraint which is (7.5,6); then polishing and varnishing at (9,4.5), then varnishing and the horizontal axis at (9,0). I defined my own colour using the PSTricks `\definecolor` command, which I named `marygold`. (I know that the flower is named *marigold*, but I named this colour after my wife, whose name is Mary.) Therefore the code includes

```
\definecolor{marygold}{cmyk}
  {0,0.1,0.5,0}
```

and

```
\pspolygon
  [fillstyle=solid,fillcolor=marygold]
  (0,0)(0,900)(750,600)(900,450)(900,0)
```

- We want to write a label next to each constraint, so that the label is parallel with the constraint. This requires recalling trigonometry from high school. For the cutting constraint, the inequality is $6X_1 + 15X_2 \leq 135$, hence the rise over the run of the boundary is -6 over 15 , or -0.4 . To find the angle that the boundary makes with the horizontal axis we need to find the arctangent of this number. PSTricks uses degrees, but Excel’s ATAN function uses radians, hence we need to use the DEGREES function as well. The Excel command is therefore `=DEGREES(ATAN(-6/15))` which is rounded to -21.801 .

A related issue is the determination of the coordinates for the centre of the expression “Cutting (1)”. For me, this was accomplished by trial-and-error. It isn’t just a matter of making the words close but not too close to the line; the words need to be placed so that they don’t interfere with other lines or words.

Each diagram done in PSTricks requires a lot of work, with many iterations to get everything right. Each iteration requires `filename.tex` \implies `dvips` \implies `ps2pdf`. Because of all the work for each diagram, I created each diagram with its own file. When I was satisfied with the final PDF, I cropped it in Adobe Acrobat, and then used an `\includegraphics` command in the main file.

3.5 Continuous improvement

Until recently, the file, now called `DecisionModeling.pdf`, was updated at least once every four months. At the outset, references were made to Excel 2010, then Excel 2013, and very recently Excel 2016. That being said, I tried to make all references to Excel to be as version-free as possible, showing the rows and columns without the ribbon.

Other improvements include: adding new examples; adding more problems to the section in each chapter on “Problems for Student Completion”; and making pedagogical improvements to the writing when students needed more explanation about a topic. Any new work is prone to typographical or other kinds of errors, so each revision provides an opportunity to make the necessary corrections.

4 Making the document open access

4.1 Introduction

My university uses a student shell called D2L (“Desire to Learn”) which is used for storing documents, submitting assignments, and storing grades. It is used always for online learning courses, and at the professor’s option for on-campus courses as well. Access to the course-specific part of D2L is restricted to those with valid course registrations. Hence, a document can be released to the students by posting it to D2L, without making the document publicly accessible.

It was obvious from the outset that this document should be made available to the students free-of-charge. It was, after all, a work-in-progress, rather than a finished product. Also, I think there would be a conflict-of-interest question in a professor adopting his own self-made publication and then collecting royalties from students. Nevertheless, during this period of development, the document contained the words “© David M. Tulett”, because I didn’t want anyone else to claim my work as their own, and I didn’t know what else I could have written in this regard. I did, however, put a note in the course outline to the effect that any student who wanted to have the document printed in a copy shop had my permission to do so.

Once the document had been completed, I had to think about whether I should try to publish it as

a commercial venture or let it be freely accessible to anyone who wanted to read it. I chose the latter, for the following reasons:

1. The content of the book was written specifically for the needs of Business 2400 at Memorial University. Other universities might consider a topic such as queueing theory, or simulation, or the study of algorithms, to be essential. Because of this, textbook publishers want manuscripts which are inclusive of all topics in the field, which is why book lengths often reach a thousand pages. To pursue this, I would have to complete much more material.
2. I am unhappy with what has happened to textbook prices. The textbook mentioned earlier now costs about \$190. There are plenty of other textbooks costing well over \$200. Most students find these prices hard to afford.
3. The existence of free material on the web helps learning.
4. Many prefer to help the public good rather than seek royalties. The \TeX community is a good example of this. I can use \LaTeX and \PSTricks without paying a fee because of the generosity of those who donated their time and effort to create and maintain these programs. In the software world, this is called “Open Source”. There are all sorts of things freely available: alternative suites to Microsoft Office; games such as chess; programming languages, and so on.

4.2 Open Access and Creative Commons licences

In a conversation with Jeannie Bail [1], a librarian at the Queen Elizabeth II Library at Memorial University, she informed me that for written material the equivalent of open source is “Open Access”. She went on to say that I should read about a major organization acting for the public good in this area, called *Creative Commons*, <http://creativecommons.org>. [2] They offer free licences which are described on their website. The most restrictive licence allows anyone to freely download the material but places the following restrictions (quoted verbatim from the Creative Commons website):

1. Attribution — You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.
2. NonCommercial — You may not use the material for commercial purposes.

3. NoDerivatives — If you remix, transform, or build upon the material, you may not distribute the modified material.

The above is a quick summary of the Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International Licence. The full text is at <http://creativecommons.org/licenses/by-nc-nd/4.0/>.

A reader of *DecisionModeling.pdf* is informed of this licence by the following graphic:



This image appears on page i, which follows the title page. Also, the word *copyright* and the copyright symbol © now no longer appear in the document.

4.3 Public availability

Memorial University encourages open access, as it helps to promote learning, and makes a website available for this purpose. For anyone who simply wants to explore what’s there, the address is <http://stor.mun.ca/>. On the website the purpose of stor is described:

stor is a learning object repository that aims to promote an atmosphere of sharing where learning objects can be searched, reused, repurposed and contributed. A learning object is a digital, open educational resource that is created to assist in a learning event. [6]

The edition of September 1 2015 was placed online at this site in March 2016. More recently *DecisionModeling.pdf* has been amended, referencing Excel 2016 instead of Excel 2013, and uploaded to stor in June 2016. [7] Here is a direct link to the current *DecisionModeling.pdf*:

<http://stor.mun.ca/handle/123456789/37463>.

With the document now in a public repository, I can inform someone of its existence by sending an email message, with a link to the document. Before this, I would have had to attach the file, which at 8 MB is rather large. I now need to make fellow educators in the field of Decision Modeling aware of the document’s existence. I thank the \TeX community for helping to make this possible.

References

- [1] Jeannie Bail, Librarian at the Queen Elizabeth II Library, Memorial University, St. John’s, NL, Canada. Personal communication, 2015.
- [2] Creative Commons, creativecommons.org, accessed June 2016.

- [3] Michel Goossens, Frank Mittelbach, Sebastian Rahtz, Denis Roegel, and Herbert Voß, *The L^AT_EX Graphics Companion*, 2nd edition, Addison-Wesley, 2008.
- [4] Memorial University, 2011–2012 *Calendar*.
http://www.mun.ca/regoff/calendar/2011_2012/sectionNo=BUSI-0288.
- [5] Barry Render, Ralph Stair, Nagraj Balakrishnan, and Brian Smith, *Managerial Decision Modeling with Spreadsheets*, 2nd Canadian Edition, Pearson Canada, Toronto, 2010. ISBN 978-0-13-208013-2, HD30.25.M35, 2010. http://wps.pearsoned.ca/ca_ph_render_mdm_2/.
- [6] <http://stor.mun.ca/>, accessed June 2016.
- [7] David Tulett. *DecisionModeling.pdf*, June 2016. Available at:
<http://stor.mun.ca/handle/123456789/37463>.
- [8] Herbert Voß, *PSTricks: Graphics and PostScript for T_EX and L^AT_EX*, UIT, Cambridge, England, 2011.

◇ David M. Tulett
Faculty of Business Administration
Memorial University
St. John's, NL, Canada, A1B 3X5
dtulett (at) mun dot ca