

December 2002

# TEACHING DSS: TRUDGING THROUGH DIRTY DATA ON THE PATH TO MODEL BASED DECISION SUPPORT

Mark Isken  
*Oakland University*

Follow this and additional works at: <http://aisel.aisnet.org/amcis2002>

---

## Recommended Citation

Isken, Mark, "TEACHING DSS: TRUDGING THROUGH DIRTY DATA ON THE PATH TO MODEL BASED DECISION SUPPORT" (2002). *AMCIS 2002 Proceedings*. 28.  
<http://aisel.aisnet.org/amcis2002/28>

This material is brought to you by the Americas Conference on Information Systems (AMCIS) at AIS Electronic Library (AISeL). It has been accepted for inclusion in AMCIS 2002 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact [elibrary@aisnet.org](mailto:elibrary@aisnet.org).

# TEACHING DSS: TRUDGING THROUGH DIRTY DATA ON THE PATH TO MODEL BASED DECISION SUPPORT

**Mark W. Isken**  
Oakland University  
isken@oakland.edu

## Abstract

*We describe an integrated set of teaching modules that allow students to explore DSS topics ranging from data warehousing to model based decision support. The modules are used as part of a spreadsheet based modeling approach to teaching DSS. We describe one particular module in detail in which students are forced to deal with a “dirty dataset” as a prelude to analysis, modeling and decision support development. Practical decision support issues are discussed and suggestions given for integrating data and model based DSS concepts in the classroom.*

**Keywords:** Decision support systems, spreadsheet, modeling, teaching, data warehousing

## Evolution of a DSS Course

The huge umbrella of decision support systems (DSS) has long provided a welcome gathering spot for those interested in building software applications based on a mixture of models, data analysis, and powerful interfaces. DSS attracts scholars and students from a range of fields including information systems, operations research/management science, computer science, finance and other business disciplines. While this breadth is a strength, it also poses challenges to those teaching DSS courses. The sheer range of potential topics poses a challenge as well as a fact that while Management Information Systems (MIS) is a strong, popular major at many business schools, the elimination of the required management science course at many universities (Powell 1998, Grossman 2001), makes it more difficult to introduce model based DSS to students at a deeper than superficial level. However, there has been a second order effect of the pressure put on management science courses – the advent of the spreadsheet based modeling course.

There has been a virtual revolution in terms of spreadsheet based management science and operations management courses which seems here to stay in business schools (Winston 1996, Powell 1998, 2001). Spreadsheets have evolved into a quite capable platform for end-user decision support modeling (Grossman 1997, Ragsdale 2001). Within Microsoft Excel, this evolution has resulted in the inclusion of Solver (Fylstra, Lasdon, Watson, Waren 1998) for optimization, Pivot Tables, data tables, database connectivity, numerous mathematical and statistical functions and the Visual Basic for Applications (VBA) programming language. Sophisticated add-ins like @Risk and Precision Tree allow analysts to do Monte-Carlo simulation and decision analysis within the familiar confines of Excel. Relational database management systems such as MS Access provide a viable platform for creating decision support tools with the inclusion of VBA, database functionality and connectivity, as well as simple yet powerful interface development tools. Taken together, these developments suggest there exists tremendous opportunity for business analysts or “end-user modelers” to pair models with rich data sources to create sophisticated decision support tools using readily available software components.

While this is all quite exciting for those with a passion for both Visual Basic and for queuing models, there are challenges involved in effective teaching of such concepts and skills in today’s business schools. At Oakland University, MIS is the largest major in the business school; there is no operations management major nor is there a management science course taught regularly. Students take six credit hours of business statistics in their sophomore year. The MIS department has offered a traditional DSS course for a number of years and in Winter 2001, the author transformed the course into a spreadsheet based DSS course. Given the heavy MIS emphasis at our school, we decided to develop a course that integrates MIS, DSS, and management science

concepts with the goal of helping students become technically adept problem solvers in practical business settings. It is taught in a computer teaching lab and is extremely hands-on. The course is designed to appeal to all quantitatively inclined business students. Having spent seven years as a practicing operations research and decision support specialist in the healthcare industry, I am absolutely convinced of the value equipping business students with this integrated tool set. While we will give a brief overview of the general nature of the course in this paper the course web site, <http://www.sba.oakland.edu/faculty/isken/mis436/index.htm>, contains detailed information about the entire course. Our primary focus will be on a particular set of modules we use to integrate concepts of data based and model based decision support. These modules were designed to provide a natural progression from the world of corporate transaction system data to data based decision support and finally to model based decision support.

## Motivation for the Teaching Modules

Many businesses have realized the tremendous value of their internal operational data for supporting decision making. This has played out in the software marketplace in the growth of the data warehousing, OLAP (online analytical processing), and data mining fields. Data warehouses provide a foundation of data that has already been merged from multiple systems, validated, transformed for purposes of facilitating analysis, stored according to a logical organizational scheme, and which is easily accessible using various client side tools such as query builders, spreadsheets and OLAP systems (Kimball 1996). Multidimensional data modeling provides an alternative way of structuring databases for business analysis (Thomsen 1997). OLAP tools such as Microsoft Excel's Pivot Table and Pivot Chart functionality provide a powerful front-end for flexible data exploration. However, no matter how good the data warehouse and data analysis tools, many common managerial decision making problems really require models to aid the decision maker in arriving at a defensible decision.

We decided to create a set of loosely integrated teaching modules that allows students to explore the inextricably linked concepts of data warehousing, OLAP, and decision modeling. In doing so, the students will also become familiar with a host of advanced Excel tools and techniques that will they will find extremely valuable in the business world. This multi-part module begins with the world of "dirty data", then on to OLAP, and concludes with queueing and optimization models. The application thread running through all three modules is that of analyzing call center data for purposes of assessing staffing capacity and customer service related questions. In this paper we will describe the first module in some detail as it is somewhat unique and briefly describe the others. We have used different combinations and versions of these modules as part of our undergraduate Applied Technology in Business (ATiB) program, our Executive MBA in Healthcare program as well as in our DSS class consisting of senior undergraduates and MBA students.

## Module 1 – Dirty Data

Perceived value of data buried in corporate information systems, the growth in enterprise resource planning (ERP) and e-commerce systems and the availability of inexpensive computing resources have all contributed to massive growth in the data warehousing and OLAP markets (Dhond et al 2000). Typically, such efforts involve a significant amount of effort to get data out of one place and into another. This phase, optimistically called extraction-transform-load (ETL), can be an unbelievably difficult exercise in dealing with missing, contradictory, ambiguous, undefined, and just plain messy data (Pyle 1999, Mallach 2000).

In general, much time is spent on getting data out of systems for:

- Analysis (spreadsheet, database, statistics package, etc.)
- Transformation and input into new system (ERP, data warehouse, etc.)

Ideally, systems can export in "nice" comma delimited, tab delimited, spreadsheet or database formats. That is not always the case with legacy systems. Often the fastest way to get data from such systems is to redirect a printed report to disk. Now the business analyst is faced with a huge electronic file that needs to be manipulated in order to make it useful. This is a necessary and important part of working with information systems. It is often dismissed as tedious yet mindless work. This is far from the truth as it provides an opportunity for clever problem solving and has direct benefit in terms of time savings.

We have created an Excel based teaching module called Dealing with Muddy Data to introduce this real business problem to our students. The module has multiple learning objectives:

1. Let students experience the difficulty of ETL activities,

2. Allow students to learn some of Excel's useful data manipulation functions such as string parsing and date/time functions and to generally raise their "power" level in Excel (which translates to most database and spreadsheet applications),
3. Help the students start to learn how to make guesses about function existence, to hunt down useful functions and to figure out how to use them from online help systems and wizards,
4. Provide students with their first look at VBA via small user defined functions to aid the ETL process,
5. Provide a bridge for our predominantly MIS audience from the comfortable world (for them) of information systems to the uncomfortable world of data analysis and modeling.
6. Cultivate an MIS mindset that abhors manual data cleaning/transformation/manipulation and is always looking for structure in data files that can be exploited to create automated solutions,
7. Remind students that knowledge work often involves getting down and dirty with data and that their value as analysts is increased by developing proficiency in handling any data set thrown at them.

There are commercial products such as Data Watch's Monarch and Data Junction's Content Extractor (formerly Cambio) (Mallach 2000) which address the general problem of extracting usable data from formatted electronic reports. However, the process of doing this task with a general purpose tool such as Excel gives students a better understanding of data manipulation concepts such as string parsing, exploiting structure in data and automation of repetitive tasks which is often required even with commercial ETL tools. Furthermore, virtually every business analyst will have access to a spreadsheet and/or database package.

### ***Structure of the Class***

We begin the session with a brief introduction to the general problem and stress the ubiquitous nature of such problems and the undesirability of brute force manual approaches due to time constraints and likelihood of error. The students are reminded that we all don't start out as strategic IS visionaries and even if we do, small companies require people who are willing to sweep the floor as well as envision the future of the Internet, in the same day. Then the specific case that we are going to use is introduced: data generated from an automatic call distributor (ACD) from a call center at a large tertiary care hospital. We take the perspective of analysts interested in studying call center performance and suggesting staffing changes that would improve customer service and/or reduce operating costs. The ACD system generated many summary reports but it was difficult to get at this data in an electronic fashion. The system was capable of sending reports to disk. So we had a large number of reports for different areas in the hospital (each area is called a "Split") and different dates generated and dumped to disk.

The students receive a tutorial handout (available from the author) which guides them through the exercise. It gives some background and then has the student open the file in a text editor such as WordPad. They get a chance to explore the structure of the file before attempting to import it into Excel. Often the default font will be a variable width font and the students immediately see the value of changing everything to a fixed width font such as Courier. An annotated snapshot of the file (Figure 1) is displayed and we discuss structural elements such as report headers and footers which contain useful information (e.g. report date), blank and other unnecessary lines, and totals sections. The notion of looking for "exploitable structure" is emphasized. For example, the Split number seems to follow the heading "Split:" and the phrase "Daily Split Report" signals the beginning of a new report.

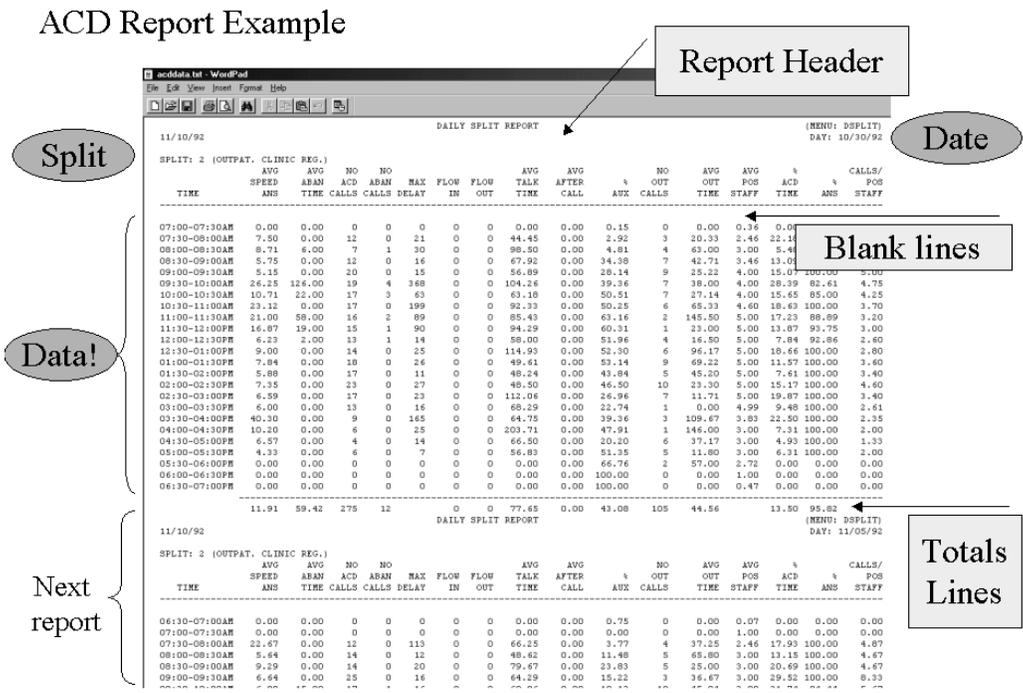


Figure 1. Sample ACD Report

### Example 1 – Single Report

We start by working with a text file that contains a single ACD report for one date. The Text Import feature of Excel, treating the file as Fixed Width (as opposed to delimited), is used to import the file. Surprisingly, most students, though regular users of Excel, had little experience even with simple text file importing. Figure 2 shows the result of this step.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	11/10/92															
2																
3																
4	SPLIT: 2 (PAT. CL. NIC REG.)															
5	AVG	AVG	NO	NO												
6	SPEED	ABAN	ACD	ABAN	MAX	FLOW	FLOW	AVG	AVG	%	NO	AVG	AVG	%		CALLS/
7	TIME	ANS	TIME	CALLS	CALLS	DELAY	IN	OUT	TALK	AFTER	OUT	POS	POS	ACD	%	POS
8																STAFF
9																
10	07:00-07:30	0	0	0	0	0	0	0	0	0	0	0	0	1.43	0	0
11	07:30-08:00	7.94	24	17	1	36	0	0	119.06	0	4.51	6	49.17	3.95	27.61	94.44
12	08:00-08:30	6	0	19	0	12	0	0	108.94	0	24.18	5	37	4	34.47	100
13	08:30-09:00	10.65	0	17	0	41	0	0	179.94	0	24.92	5	31.6	4	40.6	100
14	09:00-09:30	10.21	0	26	0	49	0	0	112.78	0	14.76	8	65.63	4	39.03	100
15	09:30-10:00	29.89	0	27	0	140	0	0	122.77	0	24.1	9	96	4	48.1	100
16	10:00-10:30	42.34	0	29	0	153	0	0	151.41	0	26.03	5	138.4	4	63.97	100
17	10:30-11:00	31.79	108.5	24	2	208	0	0	137.12	0	3.75	1	35	4	45.1	92.31
18	11:00-11:30	15.36	50.5	25	2	94	0	0	137.44	0	29.63	6	56	4	45.39	92.59
19	11:30-12:00	6.05	0	21	0	12	0	0	121.76	0	12.89	5	33.6	4	37.63	100
20	12:00-12:30	48.04	49.5	24	4	317	0	0	102.56	0	49.97	3	49.67	4	32.11	85.71
21	12:30-01:00	86.13	21	16	5	215	0	0	128.8	0	56.15	2	115.5	4	26.93	76.19
22	01:00-01:30	114.36	102.86	22	7	280	0	0	124.18	0	58.78	0	0	4	39.63	75.86
23	01:30-02:00	38.48	34.71	22	7	134	0	0	95.18	0	49.28	9	83.11	4	27.75	75.86
24	02:00-02:30	6.23	0	22	0	19	0	0	58.24	0	18.97	3	13	4	24.82	100
25	02:30-03:00	8.74	8	23	1	71	0	0	137.96	0	10.67	3	69	4	38.43	95.83
26	03:00-03:30	14.63	0	15	0	70	0	0	83.6	0	31	1	3	4	17.25	100
27	03:30-04:00	5.27	0	15	0	11	0	0	121.87	0	36.78	2	47.5	4	32.29	100
28	04:00-04:30	5.94	0	16	0	16	4	0	95.47	0	11.99	2	35	4	20.18	100
29	04:30-05:00	6.08	0	12	0	10	0	0	144.58	0	0.83	4	12.75	4	24.1	100
30	05:00-05:30	14.89	18	9	1	105	0	0	119.67	0	37.8	6	53.83	3.82	15.67	90
31	05:30-06:00	4	30	1	1	30	0	0	82	0	70.03	2	354	2	2.28	50
32	06:00-06:30	0	0	0	0	0	0	0	0	0	50	0	0	2	0	0
33	06:30-07:00	0	0	0	0	0	0	0	0	0	50	0	0	2	0	0
34	07:00-07:30	0	0	0	0	0	0	0	0	0	50	0	0	2	0	0
35	07:30-08:00	0	0	0	0	0	0	0	0	0	50	0	0	2	0	0
36	08:00-08:30	0	0	0	0	0	0	0	0	0	50.1	0	0	1.67	0	0
37																
38		26.88	53.88	404	31				121.57		29.54	87	67.61		29.38	92.87
39																

Figure 2. Result of First Import

While the numeric data in the report falls nicely into columns and rows, one important piece of information, the Split Number and Split Name have gotten somewhat garbled:

	A	B	C	D
1				
2	11/10/92			
3				
4	SPLIT: 2 (OUT	PAT. CLI	NIC REG. )	

**Figure 3. Garbled Split Name**

The students now are told that we want to create a formula in cell A3 that will return the split number, i.e. 2. They can assume that the split number is always  $\leq 100$  and they are given a hint that the Value() and Mid() functions might be useful.

Next we want to create a formula in cell J3 that will return the split name, i.e. "OUTPAT CLINIC REG." They get the following guidance:

**Step 1.** Concatenate A4:C4 with a formula in B3. Use the & operator.

**Step 2.** Find the numeric position of the left parentheses in the result in cell B3. Use the Find() function in cell F3.

**Step 3.** Use the result of Step 2 along with the Mid() function to get OUTPAT CLINIC REG. in cell G3.

	A	B	C	D	E	F	G
1							DAILY
2	11/10/1992						
3		2	SPLIT: 2 (OUTPAT. CLINIC REG.			10	OUTPAT. CLINIC REG.
4	SPLIT: 2 (OUT	PAT. CLI	NIC REG. )				
5		AVG	AVG	NO	NO		

**Figure 4. Fixing the Split Name**

Of course, several of these steps could be combined. However, students are urged to break problems like this into small formulaic chunks as it is much easier to debug multiple simple formulas than one enormous formula containing five or six nested Excel functions. After getting the desired result, one can always build a single function by copying and pasting the pieces.

### ***A Little Date/Time Problem***

It turns out that the date has luckily made it unscathed into a cell but the time periods in column A look like:

07:00-07:30AM

This is one string in a cell. The students are challenged to create formulas in two separate cells that contain the actual date/time values corresponding to the beginning and end of the period.

11/2/92 7:00 AM 11/2/92 7:30 AM

This leads to a discussion about how dates and times are stored in most spreadsheet and database packages. Our experience has been that few students are aware of how they are stored and thus cannot effectively manipulate date/time values in formulas nor take advantage of the wealth of date/time functions built into standard office productivity applications. A related discussion then is likely to ensue about the differences between date/time values, strings that look like dates, and different date/time cell formats. We also use this first example to talk about things like using the concatenation operator (&) to create unique field names in a single row and identification of the unprintable characters using the CODE() and CHAR() functions. Finally, in preparation for the second part of the tutorial and to illustrate another useful approach to text manipulation, we re-import the text file into a single column and then use the *Text to Columns* tool available in Excel to parse just the data rows into columns.

### Example 2: Many Days of Reports in One File

Now that the students have some familiarity with data importing and cleaning, we tackle a case of over sixty ACD reports from three different splits and multiple dates all contained in a single text file. The size of the file drives home the point that we really do not want to manually wade through this file to prepare it for analysis. Having spent considerable time dealing with these very problems professionally, we emphasize to the students the value of becoming proficient at quickly creating “machinery” to automate processing problems like this. Certainly, if the data files will need to be dealt with on an ongoing basis or if the file is enormous, the value of automation is clear. *However, even if the file is somewhat of a one-time obstacle and of manageable size, the value of the learning that takes place in figuring out how to automate the process usually outweighs any extra time spent doing so.* The goal is to eventually create a spreadsheet that looks like Figure 5 (many more columns to the right).

	C	D	E	F	G	H	I
1	Date	Split	Line	AVGSPEEDANS	AVGABANTIME	NOACDCALLS	NOABANCALLS
125	10/21/92	2	10:00-10:30AM	18.56	17	16	2
126	10/21/92	2	10:30-11:00AM	5.96	19	26	2
127	10/21/92	2	11:00-11:30AM	5.46	4	13	1
128	10/21/92	2	11:30-12:00PM	6.06	2	18	1
129	10/21/92	2	12:00-12:30PM	10	0	15	0
130	10/21/92	2	12:30-01:00PM	10.07	0	15	0
131	10/21/92	2	01:00-01:30PM	5.37	0	19	0
132	10/21/92	2	01:30-02:00PM	13.5	11.5	18	2
133	10/21/92	2	02:00-02:30PM	19.8	0	13	0

Figure 5. Our Goal

In other words, we’ve gotten rid of all the extraneous rows and we’ve got the report date and split number on each row so that we know where the data came from. This file is now in good shape for pivoting, importing into a database or statistics package, etc.

The tutorial walks the students through one general approach for tackling this problem. A combination of instruction, hints, screen shots, and questions are used to guide the learning experience. The students are reminded that this is certainly not the only way to proceed and they are urged to look for opportunities for other, perhaps better, approaches. Along the way they encounter concepts and tactics like:

- inserting a column containing a unique sequence number so that the original row order can be regained as needed,
- identifying unneeded rows with indicator formulas,
- use of functions like ISERROR(), ISDATE(), and ISNUMERIC() to test cells for certain conditions,
- using VBA to create user defined functions instead of nesting multiple IF() functions,
- copying and Paste-Special-Values to “freeze” intermediate results of complex expressions.

## Applications

This tutorial has never failed to elicit examples from students from their places of employment. One student, a LAN manager for an auto supplier, used these ideas to create a network utility that summarized the file content of their LAN by processing the output of a VBA automated stream of DOS DIR commands from within an Excel application. Another student was faced with a difficult data transformation problem as part of an Oracle ERP implementation. The project involved a mainframe report directed to text file with over 50,000 lines of data describing information security access levels for several thousand employees. Finally, a graduate student who was working in the University’s finance department gave us a demonstration of the commercial product Cambio and described numerous muddy data problems that they deal with routinely.

### Module 2 – OLAP with Excel Pivot Tables and Charts

The Muddy Data module is followed by a module focusing on OLAP using Excel Pivot Tables and Pivot Charts. We have found through polling our students, many who are working and are routine spreadsheet users that very few are familiar with Pivot Tables and Pivot Charts. This session begins with a discussion of basic data warehousing concepts including star schemas and multidimensional data models. We stress the differences between databases designed for analysis and databases designed to support transaction processing. For this module we have created a fictional call center that responds to technical support calls for Microsoft Office products. The application itself, computer technical support call center, was chosen partly because it falls under

the MIS managerial domain - MIS students may end up managing such call centers. The data model, implemented in MS Access, consists of time, customer, application, and problem dimensions and two facts: time spent on hold and time to service the technical support call. The dataset consists of over 26,000 rows of data generated from a simulation model of a call center. Each row is a single phone call. Like the previous module, the students are given a tutorial (available from the author) consisting of a background section and a combination of instruction, hints, screen shots, and questions.

After becoming familiar with the data model, we use the MS Access QBE (Query by Example) tool to create basic queries and quickly find that SQL is far from ideal for ad-hoc exploratory analysis. This motivates the need for flexible OLAP tools such as Pivot Tables. The students are provided with an Excel workbook containing the results of a query that combines fields from the dimension and fact tables. The bulk of the tutorial focuses on the basics of creating Pivot Tables and Charts to answer business questions such as the volume of calls by various combinations of customer, application, problem and time values. Specific topics include creating the pivot table, sections of pivot tables, pivoting, changing display types, drilling down to detail data, formatting, general table options, controlling subtotals, filtering of field values and creating pivot charts.

We recently used this tutorial with a group of healthcare executive MBA students (using data from a surgical recovery room) and it was received extremely well. They are frustrated by the lack of data to support their decision making and quickly saw the potential for these types of tools. Many of their questions focused on how to go about getting access to such data from healthcare information systems. This provided a nice lead in to the section in the tutorial where we see how to connect Excel's Pivot Table to data in an external database. Discussion of commercial OLAP tools (e.g. WebFocus) complements this topic since such tools usually are capable of working with data in a myriad of database and file formats.

As the tutorial nears the end, the students are presented with their task:

*Now that you're a bit familiar with pivot tables, let's use them to do some exploratory analysis. You are the manager of this call center. You've heard some grumbling about long waits on hold from some customers. Since you don't want to overreact to anecdotal evidence, you decide to explore the data warehouse a bit to find out more since **OnHoldMins** is a field in the database. Also, the actual staffing level of the call center is summarized by day of week and hour of day in **tblStaffLevelSummary**. That table includes a field called **WeeklyHourBin** which ranges from 1-168 and represents each hour of the week starting on Sunday at midnight. Use Excel Pivot Tables to explore this issue. Some basic questions you might want to address are:*

1. *Does overall average time on hold differ by time of day, by day of week, by both?*
2. *Do different customer types experience different hold times on average? What about by time of day and day of week?*

*As the manager, your primary way of controlling caller time on hold is through staffing and scheduling. You'd like to see a summary showing average hold time and average staff utilization by day of week and time of day. Seems like a pretty straight-forward thing for a manager of a call center to want. Turns out it's a bit of a bear to create this using SQL in Access.*

They are then challenged to create Pivot Tables and Charts which would help address these basic questions. The tutorial then ends with the following discussion and which provides the link to the next module on modeling:

*In this module, you've analyzed call center data using pivot tables and pivot charts. You've taken the perspective of the manager of the call center. As the manager, certain questions would likely arise based on the analysis.*

1. *It seems like there are times where staff utilization is really low. If we reduce staffing how will the average time on hold will be affected?*
2. *There are times when average time on hold and staff utilization are both pretty high. I bet we could reduce average time on hold by adding staff (thus reducing staff utilization). How much staff would we need to reduce average time on hold to 30 seconds? 15 seconds?*
3. *Average time on hold isn't the only important performance measure. We're also interested in the percentage of customers who have to hold and the percentage of customers who hold less than 1 minute.*

*How could we calculate these measures from the current data and how could we figure out how they will be affected by changes in staff levels?*

4. *We currently just have our tech support staff working 8-hour shifts. The midnight shift start is from 12a-8a, the day shift is from 8a-4p, and the afternoon shift is from 4p-12a. If we were better able to estimate our staffing needs by hour of day and day of week, how would we figure out how to schedule our staff (e.g. shift start times and shift lengths) to meet these staffing needs?*

*These are not easy questions but are the type routinely faced by managers of a myriad of customer service related systems including software tech support, retail sales (L.L Bean, Lands End), healthcare, police and emergency services, fast food, insurance companies, and product support. While historical data is useful for analyzing current system performance, we need something else to handle “What if?” or “What’s best?” type questions. We need models. Try to think a bit about what makes questions 1-4 difficult and we’ll discuss modeling with spreadsheets next time.*

## **Additional Modules – Queueing and Optimization Models for Call Center Staffing**

The previous two modules have set the stage to discuss the use of queueing models to explore the relationship between capacity and system performance and customer service measures such as average and upper percentiles of time on hold, employee utilization, and lost calls. We use the spreadsheet modeling based textbook, “Practical Management Science” (Winston and Albright, 2001) which includes a very nice chapter on basic queueing models. The students also receive a tutorial which introduces queueing models in the context of our call center example. The basics physics of the M/M/s queue (exponential interarrival times, exponential service times and s servers) are discussed and the students work with a pre-developed spreadsheet model that calculates performance measures for user defined parameter values such as the call arrival rate and service time. The objective is to have students discover that, 1) retrospective data analysis is simply not sufficient to answer these kinds of basic operational questions, 2) mathematical models can provide a way to address such questions, 3) models are inherently based on a host of simplifying assumptions and 4) they should have paid better attention in their math and probability/statistics courses.

The last objective is a real issue (Groleau 1999). It is valuable for students to see “what’s going on underneath the hood” of these mysterious things called mathematical models. However, the mathematical background of many of the students is sparse and care must be taken to present the material so they do actually gain insight from it. By exploring the queue length and wait time distribution formulas for the M/M/s queue, students see that 1) everything in business cannot be described by linear regression lines, 2) very powerful and informative statements about real business systems can be made by equations containing nothing more complex than summations, exponentiation, and factorials.

Then students answer basic questions using the M/M/s spreadsheet model concerning the relationship between wait time and changes in call volume, staffing to meet service level goals, and use of Little’s Law. They also are forced to think about how doing iterative calculations in a spreadsheet and the relative merits of cell based formulas and procedures written in VBA. This brings MIS development issues back into a discussion about modeling.

Finally, we discuss the fact that though we can use this queueing model to estimate staffing needs by time of day and day of week, our job is far from over. Now we’ve got to figure out how to actually develop work schedules that meet our staffing targets and conform to policies concerning shifts lengths, start time, days worked per week, and other employee desires and constraints concerning schedules. This provides a way to introduce optimization models using Solver in Excel. Over several class sessions, the students are introduced to the basics of linear and integer programming using Solver. They explore and enhance a basic employee scheduling model. We then use this model to provide a bridge to talking about creating full blown decision support systems on the Excel platform, complete with a graphical user interface that hides complex mathematical models from the user (see Zobel, Ragsdale and Rees 2000, Ragsdale 2001).

## **Summary**

We have developed a number of related modules that take students on a journey through a range of data and model based DSS topics. Since our class is heavily MIS dominated, we begin with data cleansing and OLAP tutorials based on data from a computer technical support call center. These tutorials naturally lead the students to difficult managerial questions involving call center staffing and scheduling for which mathematical models prove extremely helpful. In general, students have responded very

favorably to this approach and a number of commented that they were immediately able to apply many of the things they learned at their places of employment. The greatest pedagogical challenge has been to structure the course so as to move at a pace the students can handle and yet cover the range of topics necessary for the students to actually build Excel based DSS's for their end of the term project. In addition to the topics discussed above, students are taught to program in VBA (using Albright 2001), develop basic user forms that communicate with their spreadsheet models, do spreadsheet simulation with @Risk, and decision analysis with Precision Tree. Obviously, in a single semester, we can only explore each topic to a limited degree. Most students leave the class as "power users" of Excel and with a newfound appreciation for the power lurking in this unassuming tool. The course has now been renamed Business Analysis and Modeling and is paired with a Decision Support Systems course. This allows us to better introduce a wide range of DSS topics in greater depth than we could with a single course.

## References

- Albright, S.C. "VBA for Modelers", Duxbury, 2001, Pacific Grove, CA.
- Dhond, A., Gupta, A., and Vadhavkar, S. "Data Mining Techniques for Optimizing Inventories for Electronic Commerce", Proceedings of KDD 2000, 2000, Boston, MA.
- Foulds, L. and Thachenkary, C. "Empower to the People: How Decision Support Systems and IT Can Aid the Analyst", *OR/MS Today* (28:3), June 2001, pp. 32-37.
- Fylstra, D., L. Lasdon, J. Watson, A. Waren, "Design and use of the Microsoft Excel Solver", *Interfaces*, (28:5), Sept.-Oct. 1998, pp. 29-55.
- Groleau, T.G. "Spreadsheets Will Not Save OR/MS!", *OR/MS Today*, February 1999.
- Grossman, T. A. Jr. "End User Modeling", *OR/MS Today*, October 1997.
- Grossman, T.A. Jr. "Causes of the Decline of the Business School Management Science Course," *INFORMS Transactions on Education*, (1:2), June 2001, <http://ite.informs.org/Vol1No2/Grossman/Grossman.html>.
- Kimball, R. "The Data Warehouse Toolkit, John Wiley & Sons, Inc., 1996, New York.
- Mallach, E.G. "Decision Support and Data Warehouse Systems", McGraw-Hill, 2000.
- Powell, S.G., "The Teachers' Forum: From Intelligent Consumer to Active Modeler, Two MBA Success Stories?", *Interfaces* (27:3), May-June 1997, pp. 88-98.
- Powell, S.G., "The Teachers' Forum: Requiem for the Management Science Course?", *Interfaces* (28:2), March-April 1998, pp. 111-117.
- Powell, S.G., "Teaching Modeling in Management Science?", *INFORMS Transactions on Education* (1:2), June 2001, <http://ite.informs.org/Vol1No2/powell/powell.html>.
- Pyle, D. "Data Preparation for Data Mining", Academic Press, 1999, San Diego, CA.
- Ragsdale, C.T. "Teaching Management Science With Spreadsheets: From Decision Models to Decision Support", *INFORMS Transactions on Education* (1:2), June 2001, <http://ite.informs.org/Vol1No2/Ragsdale/Ragsdale.html>
- Thomsen, E. "OLAP Solutions: Building Multidimensional Information Systems", Wiley Computer Publishing, 1997, New York.
- Winston, W. "The Teachers' Forum: Management Science with Spreadsheets for MBAs at Indiana University?", *Interfaces* (26:2), March-April 1996, pp. 105-111.
- Winston, W. "Executive Education Opportunities: Millions of Analysts Need Training in Spreadsheet Modeling, Optimization, Monte Carlo Simulation and Data Analysis", *OR/MS Today*, August 2001.
- Winston, W. and Albright, S.C. "Practical Management Science", Duxbury, 2001, Pacific Grove, CA.
- Zobel, C., Ragsdale, C.T., and Rees, L. "Hands-on Learning Experience: Teaching Decision-Support Systems Using VBA", *OR/MS Today*, December 2000.