Proceedings of the 2005 Winter Simulation Conference
M. E. Kuhl, N. M. Steiger, F. B. Armstrong, and J. A. Joines, eds.

COMBINING HANDS-ON, SPREADSHEET AND DISCRETE EVENT SIMULATION TO TEACH SUPPLY CHAIN MANAGEMENT

Jeffrey Adams
Jerry Flatto
Leslie Gardner

School of Business
1400 East Hanna Avenue
University of Indianapolis
Indianapolis, IN 46227 U.S.A

ABSTRACT

This paper describes the effect of combining hands-on simulation with spreadsheets and discrete event simulations. These tools enhance the student learning process of supply chain management principles. Active, hands-on learning is one of the most effective types of learning but is very time consuming. Supplementing it with computer simulation enhances the hands-on learning to cover more material in less time making an efficient and effective learning experience.

1 INTRODUCTION

Active learning, where students are involved in hands-on activities, discussion, teamwork, and problem solving, is widely accepted as a far more effective mode of instruction than lecture alone (Chickering and Gamson 1987; Prince 2004). Games and hands-on simulations have become popular vehicles for active learning in business education and are commercially available as board games and in computerized versions on the Internet from a variety of sources.

The Just-In-Time (JIT) simulation and a Beer Distribution Game (beer game) are the two games that are integral parts of the operations management courses at the University of Indianapolis. The JIT simulation simulates the operation of a manufacturing system using different operating strategies with regard to transfer batch sizes and to the reduction in variation of the amount of product processed. The authors learned this game at a meeting of the Central Indiana Chapter of the American Production and Inventory Society several years ago but have been unable to trace its source. The Beer Distribution Game (Sterman 1989; Chen and Samroengraja 2000; Jacobs 2000) is a popular classroom exercise for demonstrating material and information flows in a supply chain that was developed by the Systems Dynamics Group at the Massachusetts Institute of Technology's Sloan School of Management. It dramatically impresses students with the bullwhip effect (Chatfield et al. 2004; Warburton 2004). The bullwhip effect is the amplification of variance in demand up the supply chain resulting first in large backlogs and then in excessive inventories. Charts of inventory levels have a shape similar to that of a whip when it is cracked.

The problem with learning games at the University of Indianapolis is that the course schedules permit one class period for the JIT simulation and one class period for the beer game. The JIT simulation is the perfect vehicle for teaching the benefits of various operating strategies and the value of stochastic simulation as a tool to learn how systems behave. Students could learn much more from the beer game if time permitted them to experiment with ordering strategies in order to learn which strategies are most effective. Unfortunately, time does not permit the faculty to reinforce the concept that repetition of simulation is necessary to assure that the observed behavior of the system is typical behavior. To overcome these difficulties, the operations management faculty at the University of Indianapolis has developed spreadsheet and discrete event simulations to integrate with the hands-on games that enhance students' learning. This paper covers the games and simulations in the order that they appear in the operations management course.

2 THE JUST IN TIME SIMULATION

The Just-In-Time (JIT) Simulation is a hands-on classroom activity using poker chips and dice that simulates the material flow in a manufacturing system. Students are seated at a long table, each having a place mat as illustrated in Figure 1, with four poker chips on their incoming dock, and a die.

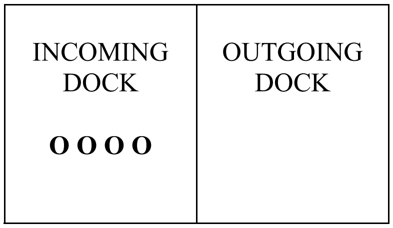

Figure 1: Placemat for Participant in JIT Simulation Game

Each student represents a worker in an assembly line, the poker chips are the product he/she is working on, and the die is used to introduce randomness into the operation of the system. The first student in line is the receiving department and the last student is the shipping department.

The students run the assembly line for a simulated month under each of three different operating strategies. A simulated month consists of 20 dice rolls with students moving poker chips after each roll of the die to simulate material flows. The students roll their dice 20 times to represent twenty working days in a month.

Under the first operating strategy, students roll the die and move the number of chips from the incoming to outgoing dock indicated by the die. For example, if the student rolls a three, he/she moves three chips from the incoming to the outgoing dock. If the student has fewer chips on the incoming dock than the number rolled on the die, the student moves all of the chips to the outgoing dock. This means that if the student rolls a six but only has four chips on the incoming dock, he/she moves all four chips to the outgoing dock. After the students move the chips to their outgoing docks, they may pass on transfer batches of four chips to the next student's incoming dock.

The second operating strategy is identical to the first except that the transfer batch size is changed to 1, that is, the students may move all of the chips on their outgoing dock to the next student's incoming dock.

The third operating strategy involves variance reduction. If a student rolls a one, two, or three on the die, he/she moves three chips from the incoming to the outgoing dock. If a student rolls a four, five, or six on the die, he/she moves four chips from the incoming to the outgoing dock. The transfer batch remains the same.

Under each operating strategy, each student processes 3.5 chips on average because that is the average of the numbers that appear on a die. The theoretical output of the system is a total of 70 chips, that is, 3.5 per simulated day over a period of 20 days. Under the first strategy, students rarely achieve an output of more than 50 and it is often much lower. Under the second strategy, they usually produce slightly more than under the first strategy, but almost never more than the upper 50s. Under the third strategy, the students usually come very close to achieving the goal of 70. The most dramatic results are achieved if at least ten students are in an assembly line.

The JIT simulation enables the instructor to use humor to make a variety of points about good management. If time permits, the instructor can do variations where chips are removed to simulate defectives and the impact that defectives have on output. Most importantly, the students see realistic behavior of a system and grasp the concept that behavior can be modeled by an abstraction of the system that includes randomness. What it lacks is speed in obtaining results because each simulated month typically takes about 20 minutes to do. Unfortunately, students often do not believe the results because they believe its cause is the randomness of the dice, not the operating strategy. Limitations of class time prevent replication, but using a computer model for the replications overcomes the time problem.

2.1 Spreadsheet Simulation of the JIT Simulation

The primary goal of the JIT simulation is to teach students the differences between operating strategies in manufacturing systems. The secondary goal is to teach the value of stochastic simulation. The hands-on JIT simulation is reproduced in Microsoft EXCEL:

- To demonstrate that computer models can simulate the behavior of real world systems,
- To show how much faster a computer can obtain the same results as a hands-on demonstration, and
- To do many repetitions of a stochastic simulation to impress upon students that, although there is some variation from run to run, results of the three operating strategies can be replicated.

Once the spreadsheet is set up, replications of the simulation can be generated with a keystroke as opposed to 20 minutes for the hands-on version. Students can look at the throughput values for several replications of each operating strategy in a matter of seconds.

The first four days of the spreadsheet simulation for the first operating strategy are shown in Table 1. Workstations are represented by rows. The number of chips on the incoming dock is maintained in the *in* column and the number of chips on the outgoing dock is maintained in the *out* column. Four more chips are added each day to the *in* cell of *wkstn*1. The *rnd* column gives the roll of the die and uses the RANDBETWEEN function. The roll of the die gives the number of chips moved from the incoming to the outgoing dock subject to the limitation that no more chips can be moved than the number on the incoming dock. If at least four chips are on the outgoing dock, then batches of four are transferred to the next workstation for the next day. In Table 1, no chips are transferred from workstation 1 on day 1 to workstation 2 on day 2 because less than four chips are on the outgoing dock. However, eight chips (two batches of four) are transferred from workstation 1 on day

3 to workstation 2 on day 4. The average throughput for 50 replications with a summary of the operating strategies is given in Table 2.

Table 1: The First Four Days of the Spreadsheet Version of the JIT Simulation

	Day 1			**Day 2**			**Day 3**			**Day 4**		
	in	**rnd**	**out**	**in**	**rnd**	**out**	**in**	**rnd**	**out**	**in**	**rnd**	**out**
wkstn1	4	1	1	7	1	2	10	6	8	8	2	2
wkstn2	4	5	4	0	4	0	0	1	0	8	1	1
wkstn3	4	4	4	4	2	2	2	4	4	0	1	0
wkstn4	4	5	4	4	3	3	1	6	4	4	5	4
wkstn5	4	2	2	6	3	5	3	5	4	4	4	4
wkstn6	4	1	1	3	1	2	6	5	7	5	1	4
wkstn7	4	4	4	0	2	0	0	5	0	4	4	4
wkstn8	4	2	2	6	2	4	4	1	1	3	5	4
wkstn9	4	6	4	0	1	0	4	2	2	2	3	4
wkstn10	4	5	4	4	4	4	0	3	0	0	4	0

Table 2: Summary of Operating Strategies and Throughputs for 50 Replications of the Spreadsheet Simulation

Operating Strategy	Chips Processed	Batch Size	Average Throughput
1	1-6	4	41.20
2	1-6	1	52.14
3	3-4	1	66.72

Following the hands-on simulation, the instructor e-mails the students the spreadsheet simulation for them to experiment on. The students send back a response describing their experiences with the hands-on simulation, the spreadsheet simulation, what they learned from each and whether the results of the hands-on and spreadsheet simulations are consistent.

2.2 Computer Simulation Using Extend

Another approach that can be used to teach students the impact various ordering strategies have in manufacturing systems is through the use of computer simulation. One such simulation used at the University of Indianapolis is a software package called Extend <http://www.imaginethatinc.com>.

Extend is a powerful object-oriented computer simulation package but one that is also easy to use by students. The basis of the package is a "block". Each block simulates a specific role such as a queue, machine, labor resource pool, etc. Extend comes with hundreds of blocks grouped into libraries such as "Generic", "Discrete Event" or "Manufacturing" and then arranged in the libraries by categories such as "queues", "activities", and "routing". Developing the JIT model for Extend was a relatively simple process. The heart of the model consists of six blocks shown below in Figure 2. These blocks correspond to the activities performed by a single student in the hands-on portion of the exercise.

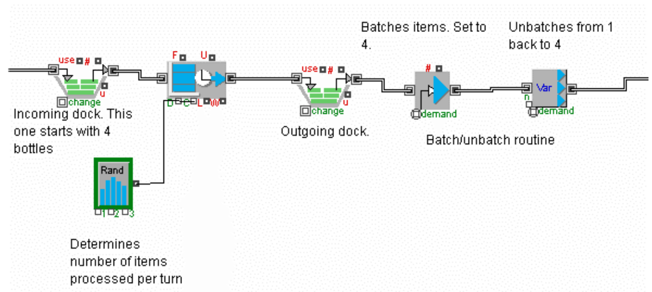

Figure 2: The JIT Model in Extend

The first block is a queue block corresponding to the incoming dock. The next block corresponds to the activity of moving chips from the incoming dock to the outgoing dock (which is the next block). The next block handles batching the items into groups of four for release while the next block ungroups the single group back into four separate items. While this last piece may seem an awkward method to handle the batching and unbatching process, the approach is easily understandable to students and also provides flexibility for a wide variety of batching and unbatching strategies.

The last block is used to determine how many chips should be processed every week. The way the block is linked via a connector to the activity block overrides the values set in the dialog box of the activity block.

Figure 3, Figure 4, and Figure 5 show a couple of typical dialog boxes from the model. Figure 3 shows the dialog box associated with the queue. The user can set an initial value for the queue. In this case, the initial value for the queue is set at four. Figure 4 shows the activity dialog box. The user can set the delay as well as the maximum number the activity can process per unit time. This last value is overridden by the distribution set in Figure 5. While not shown, the other blocks used in the model have similarly easy to use and modify dialog block.

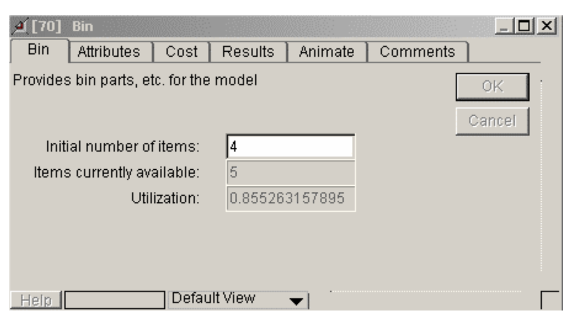

Figure 3: The Queue's Dialog Box

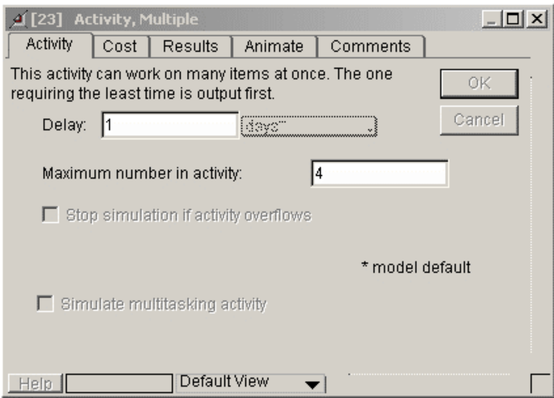

Figure 4: The Activity's Dialog Box

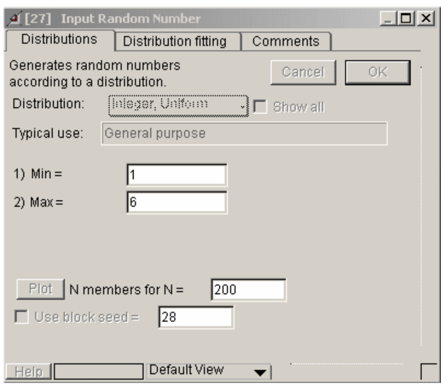

Figure 5: The Distribution Dialog Box

To simulate a group of students, these six blocks simply need to be copied and pasted and connected together. Once the model of one student was created, it took less than five minutes to create the model for twenty students to match the Excel Spreadsheet model. To increase the model to forty students in length could be accomplished with a single cut and paste in less than one minute.

Once the first model was created that simulated a die role of one through six with an associated batch size of four, creating the other two models literally took less than five minutes each. The second model simply required that the batch size be changed from four down to one. This required changing two dialog boxes in the Extend model and then copying the modified blocks as needed. The third model simply required changing the distribution from one to six down to outputting a value of three or four to simulate reduced variance.

Thus one of the features that Extend brings to being used in a classroom environment is the ease of modification of the simulation just by changing some values in the dialog box. If desired, students could easily make a wide variety of changes to the model in terms of batch sizes, activity delays, distributions, etc. and see their effects in real time.

Another useful feature is the ability of Extend to animate the model during the run process. With animation selected, the model execution is slowed considerably so that students can watch the flow of a variety of different shaped icons through the model. During the model's run, students can also view dialog boxes to monitor various values such as queue lengths.

During the simulation set up process, the user can set the start and end times, the number of simulation runs and identify any time units to be used. Table 3 shows the results when running the Extend model for sixty runs.

Table 3: Summary of Operating Strategies and Throughputs of the Extend Simulation

Operating Strategy	Chips Processed	Batch Size	Average Throughput
1	1-6	4	42.96
2	1-6	1	56.38
3	3-4	1	59.84

Following the hands-on and spreadsheet simulation, the instructor e-mails the students the Extend simulation for them to experiment on. The students send back a response describing their experiences with the hands-on simulation, the spreadsheet simulation and the Extend simulation, what they learned from each and whether the results of the hands-on, spreadsheet and Extend simulations are consistent.

2.3 Student Reactions

Students thoroughly enjoy the hands-on simulation. The spreadsheet simulation is not nearly so much fun, but the spreadsheet does reinforce the point that variance reduction is the key to productivity.

Like the hands-on simulation, the spreadsheet is understandable but not really attention-getting. The spreadsheet provides a good transition to the concept of simulations and what is happening inside the simulation process. The spreadsheet in its current form is not visually appealing and this may be improved in the future to make it more intuitive.

The Extend simulation is more visually appealing and understandable to the students than the spreadsheet. Since Extend supports animation, students can watch the flow of items through the spreadsheet. Students can also adjust various parameters such as the distribution for the number of chips processed as well as the batching size to see the impact the changes have on the overall throughput.

3 THE BEER GAME

At the University of Indianapolis, the beer game is introduced as a board game with four players: a retailer, a wholesaler, a distributor, and a factory. Customer orders are placed with the retailer who fills them to the extent possible. The retailer then orders from the wholesaler to replenish his/her stock. Similarly the wholesaler fills retailer orders and replenishes from the distributor who in turn fills wholesaler orders and replenishes from the factory. The factory fills distributor orders and replenishes from a limitless supply of raw material. All players keep records of backlogs, or unfilled orders, and attempt to fill them as soon as possible. Shipping delays of two periods separate each player, as do information delays of two periods. Initially, all four players have twelve units of inventory and four units of inventory are on each square representing a shipping delay. Similarly, all of the orders in the information pipeline at the start of the game are for four units. The game board is shown in Figure 6.

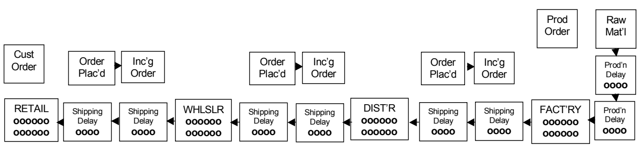

Figure 6: The Initial Setup for the Board Game Version of the Beer Game

The goal of the game is to fill all customer orders without carrying excessive inventories or having excessive backlogs. The players must fill backlogs eventually. For the first several periods of the game, the customer orders are at four units each period. At some point, the customer orders jump to eight units and remain at that level for the rest of the game. The game runs for 50 periods or until the players become frustrated with excessive backlogs and inventories and the point about the bullwhip effect has been made.

The only stochastic part of the beer game is the human behavior in placing orders. Sterman (1989) modeled this behavior based on the results of 48 trials (192 subjects) over four years at MIT's Sloan School of Management and hypothesized that the bullwhip effect, at least in the beer game, is caused by human behavior. The goal of creating simulations for classroom demonstration is to teach students ordering behavior that does not produce the bullwhip effect in the beer game. The simulations are necessary because the beer game takes between an hour and an hour and a half to play – a full class period, and to demonstrate 6-10 ordering strategies by playing the beer game would take 6-10 additional class periods which cannot be accommodated due to the necessity of covering other material in the syllabus.

3.1 Spreadsheet Simulation of the Beer Game

The Beer Game is simulated in Microsoft EXCEL to generate charts of inventory and backlog for classroom lecture purposes to demonstrate the impact of various ordering strategies in filling backlog and maintaining low levels of inventory. The ordering strategies tested in the spreadsheet simulation are:

- Naïve forecast – order only what is ordered by the immediately downstream player
- Exponential smoothing at α=0.1, 0.5, and 0.9
- Order what is ordered plus cumulative backlog
- Order what is ordered plus cumulative backlog unless inventory is some amount more than order plus cumulative backlog
- Order what is ordered plus this period backlog (not cumulative)
- Point of sale (POS) data used by all players

The spreadsheet itself is too complicated for most undergraduate operations management students to create and manipulate on their own. Since these are deterministic simulations, students need not experiment with repetitions.

The Microsoft EXCEL simulation sets up each period, or simulated week, of the beer game in three steps in the spreadsheet as shown in Table 4. Ordering strategies are represented as a formula in the *in* column for each player on the info line of the first step (*place/adv ord*) for each week. The rest of the spreadsheet consists of simple formulas referencing cells in the step before to move material and orders, to add and remove material from inventory, and to track cumulative backlog. In the spreadsheet simulation, all players use the same ordering strategy. It is important to notice that the spreadsheet simulations of the beer game are entirely deterministic. The stochastic element of the players' choice of amount to order has been removed and replaced with a formula for the ordering strategy.

Table 4: Spreadsheet for One Time Period (Simulated Week) of the Beer Game

Week	**2**											
place/adv ord	**out**	**inv**	**in**	**out**	**inv**	**in**	**out**	**inv**	**in**	**out**	**inv**	**in**
info	4	**retail**	4	4	**whol**	4	4	**dist**	4	4	**mnfr**	4
mat'l	4	12	4	4	12	4	4	12	4	4	12	4
backlog	0			0			0			0		
advmatl	**out**	**inv**	**in**	**out**	**inv**	**in**	**out**	**inv**	**in**	**out**	**inv**	**in**
info	4	**retail**	4	4	**whol**	4	4	**dist**	4	4	**mnfr**	4
mat'l	0	16	4	0	16	4	0	16	4	0	16	0
backlog	0			0			0			0		
fillord	**out**	**inv**	**in**	**out**	**inv**	**in**	**out**	**inv**	**in**	**out**	**inv**	**in**
info	0	**retail**	4	0	**whol**	4	0	**dist**	4	0	**mnfr**	4
mat'l	4	12	4	4	12	4	4	12	4	4	12	0
backlog	0			0			0			0		

The first strategy tested is the naïve forecast. A player using the naïve forecast simply orders what the immediately downstream player just ordered. In other words, if the downstream player ordered four units, this player orders four units. This strategy produces no bullwhip effect, but the retailer, wholesaler, and distributor never fill their backlog although the backlog remains steady as shown in Figure 7.

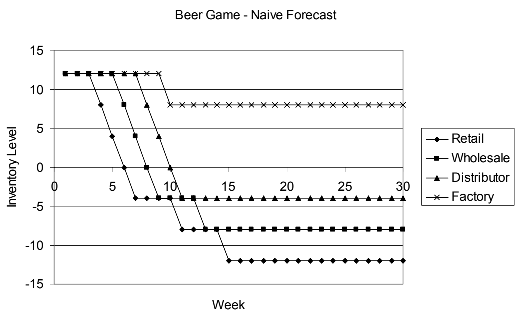

Figure 7: Results of Using the Naïve Forecast as an Ordering Strategy

Exponential smoothing, as an ordering strategy, is tested for α=0.1, 0.2, . . . , 0.9. Exponential smoothing forecasts according to the formula:

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

where

- F_t is this period's forecast
- A_{t-1} is last period's actual order
- F_{t-1} is last period's forecast.

For example, if the last order was 8 and the last forecast was 4:

- If α=0.1, F_t =(0.1)(8)+(0.9)(4)=4.4 (rounds to 4).
- If α=0.5, F_t =(0.5)(8)+(0.5)(4)=6.
- If α=0.9, F_t =(0.9)(8)+(0.1)(4)=7.6 (rounds to 8).

The results for α=0.1, 0.2, 0.3, and 0.4 are similar to those for α=0.1 shown in Figure 8 where the retailer's backlog increases without ever leveling off but the other players are able to maintain a reasonable level of inventory. The results for α=0.5, 0.6, 0.7, 0.8 and 0.9 are similar to those for α=0.5 shown in Figure 9 where all inventories initially drop but backlogs ultimately stabilize.

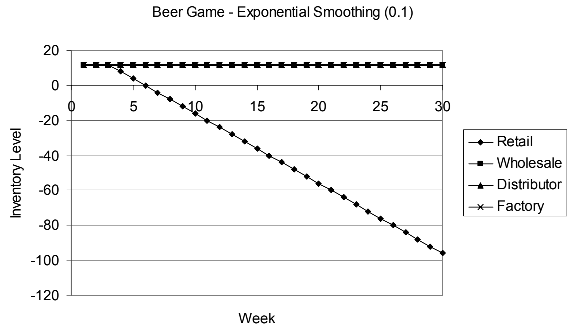

Figure 8: Results of Exponential Smoothing as an Ordering Strategy with α=0.1

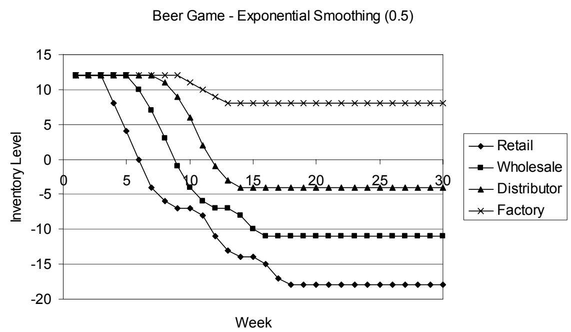

Figure 9: Results of Exponential Smoothing as an Ordering Strategy with α=0.5

Ordering strategies such as the naïve forecast and exponential smoothing that do not compensate for backlog do not create a bullwhip effect but never catch up with the backlog either. In reality, people tend to underestimate what is in the supply line (Sterman, 1989) and order based on cumulative backlog. For example, if a player had 12 ordered last period, has 8 in stock, and a cumulative backlog of 12, he or she places an order of 12+12=24 to compensate for the backlog. The player ships 8, and now has a backlog of 16. The increased order size ripples up the supply chain. The results of players ordering what is ordered plus cumulative backlog are shown in Figure 10, which dramatically displays the bullwhip effect.

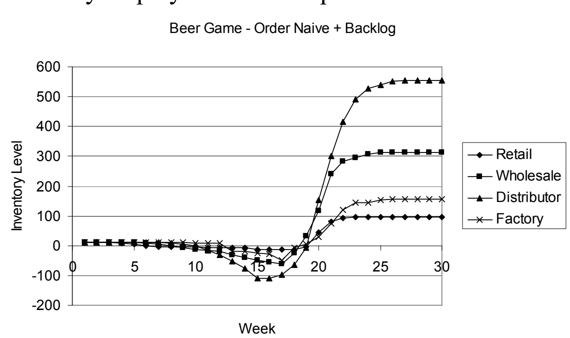

Figure 10: Results of Ordering What Was Ordered Plus Cumulative Backlog

The problem with ordering what was ordered plus cumulative backlog is that at some point, people know that they have much more in inventory than the customers are ordering and they quit ordering from upstream players. Several strategies of when to quit ordering were tested with similar results to those shown in Figure 10, except that the inventory levels for each player eventually return to the starting levels at about week 150.

Overcompensation for backlog certainly causes the bullwhip effect. If overcompensation is due to underestimation of what is in the pipeline, then disciplined ordering of only the unfilled orders for the current period, that is, non-cumulative backlog should prevent the bullwhip effect. Figure 11 shows that this is not quite the case. The time lag causes a dip in inventory levels that return to close to the original levels.

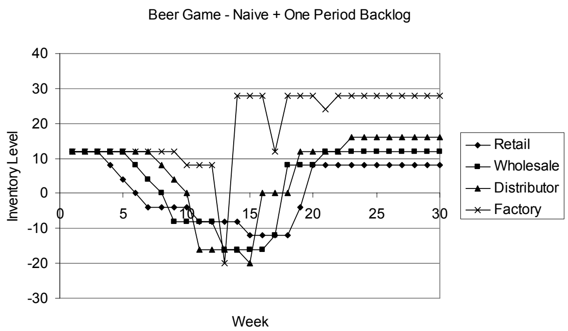

Figure 11: Results of Ordering what was Ordered Plus a Single Period of Backlog

Using point of sale data for orders up and down the supply chain is the widely touted solution to the bullwhip effect (Lee et al. 1997). This means that if *x* units are ordered from the retailer, this information is passed to all other players and all players order *x* units. This does not produce a bullwhip effect but inventory levels do not stabilize at the starting levels as shown in Figure 12.

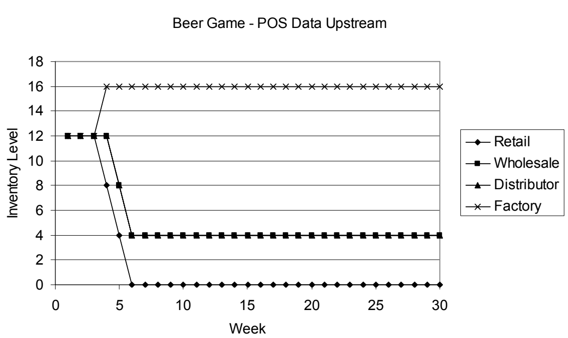

Figure 12: Results of Point of Sale Amounts Up Supply Chain

In summary, a player must compensate sufficiently for backlog or fall further and further behind. Overcompensation for backlog causes the bullwhip effect. The key to playing the beer game well is finding the right amount of compensation. These charts supplement the lecture following the beer game. To reinforce this lesson, students are encouraged to play versions of the beer game on the Internet and report how they did to the class.

3.2 The Beer Simulation Using Extend

Similar to the Just in Time simulation discussed previously, an Extend simulation was also created for the Beer game. This model was more difficult to create because of the necessity to pass information, specifically order size, backwards in the simulation. This was needed since the order sizes were determined based on downstream orders. In other words, the number of cases of beer bought by the customer determined the number of cases of beer ordered by the retailer from his/her wholesaler. From a simulation and modeling perspective, the wholesaler shipped cases of beer downstream to the retailer while receiving order sizes upstream from the retailer.

Complicating the modeling effort was the series of delays associated with each step in the overall process. Thus in addition to varying order sizes, there were also timing issues associated with the order sizes.

The solution to the problem was found via the use of Extend's capability of associating attributes with items passing through the simulation. An attribute was created called "order size" with this attribute being associated with each case of beer generated.

A piece of the Extend model is shown in Figure 13. This section models the distributor's plant as well as the two delay cycles associated with the plant. The vast majority of the model is composed of a queue and an activity linked together. This combined building block can then be used as a distributor, wholesaler, delay, etc.

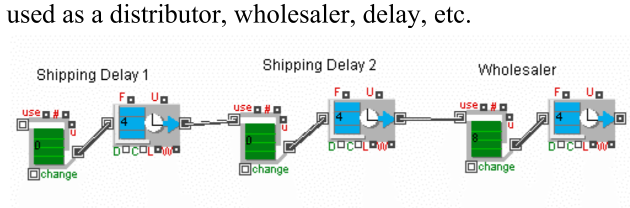

Figure 13: The Beer Model in Extend

Extend provides a variety of methods to input data into the simulation. Information can be entered from an external file or from a block on the spreadsheet where the user can directly enter the order sizes for each week. Figure 14 shows the dialog box where the customer order size is stored. Extend also supports a variety of mathematical and decision functions that can be used to modify the various order sizes to account for exponential smoothing, adding backlogs to orders, decisions on the size of the order, etc.

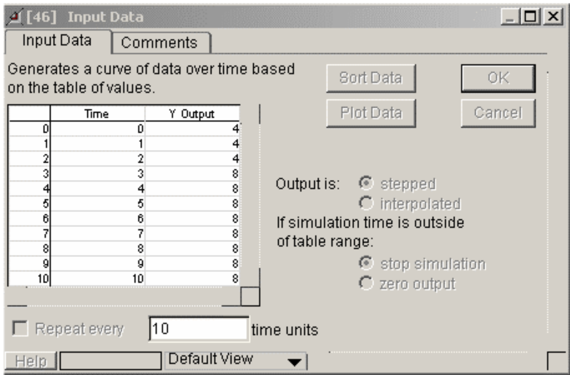

Figure 14: Customer Order Size

As mentioned previously, to handle the information concerning order size, which goes backward in the simulation, an “order size” attribute was associated with each case of beer. Using “set attribute” and “get attribute” blocks in Extend, the customer order size is then sent back to the retailer for their order size, which then is sent back to the wholesaler, and so on. Figure 15 shows the feedback of customer order size backwards in the model.

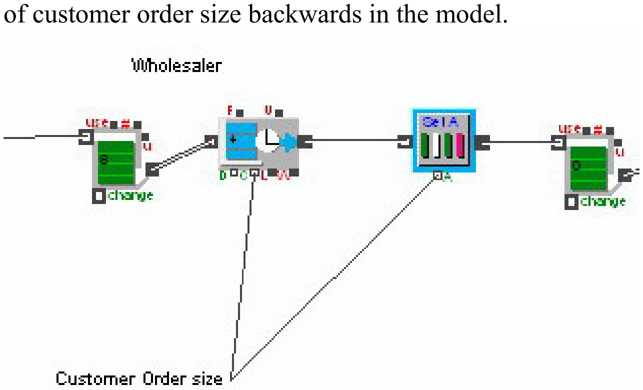

Figure 15: Feeding Back the Customer Order Size

Since the Beer model is deterministic, running the Extend model for multiple times provides no added benefit. Using the existing features and blocks in Extend, it was a relatively simple process to handle the various ordering strategies discussed in the spreadsheet model.

3.3 Student Reactions

The beer game as a board game never fails to produce the bullwhip effect and a lot of frustration for the students. The frustration opens their minds to considering the strategies tested by the deterministic spreadsheet simulation and presented to them in class.

The spreadsheet is not as intuitive as the spreadsheet for the JIT game. The Beer model is more complicated since there is two-way flow. The materials flow downstream while the order size information flows upstream. Therefore the spreadsheet does not lend itself to having students experiment with the spreadsheet itself. Thus the model is primarily useful to supplement the lectures.

On the other hand, the Extend model is more understandable because of the visual nature of the model as well as the animation and the ability to view the order size information as the model runs. Using the dialog boxes, students can also easily change ordering strategies to see their impact on the bullwhip effect.

As with the Internet simulations of the beer game, students are expected to adjust ordering strategies to see their effect. The results of the hands-on. Spreadsheet, Internet and Extend simulations are then discussed in class.

4 BENEFITS OF COMBINING HANDS-ON, SPREADSHEET, AND DISCRETE EVENT SIMULATION

The concrete experience of a hands-on game followed by spreadsheet or discrete event simulations of the game allows students to grasp more quickly seeing the game on a computer. Animations for discrete event simulations are more intuitive for students, but randomness is more visible in the spreadsheet where they can click on a cell and see the RAND or RANDBETWEEN function. Simulations allow us to demonstrate scenarios and repetitions quickly that time does not allow us to do with the hands on version in a classroom setting.

Students can easily change various parameters of the Extend simulations using the dialog boxes so students can learn from experimentation with the models. The built-in animation support provided by Extend allows the students to better follow the flow of materials and information through the model. Additionally, Extend provides various graphing and statistical analysis blocks to perform further analysis upon the information provided by the simulation as well as allowing the export of information to a spreadsheet program.

Active, hands-on learning is one of the most effective types of learning but is very time consuming. Supplementing it with computer simulation enhances the hands-on learning to cover more material in less time making an efficient and effective learning experience.

REFERENCES

Chatfield, D. C., J. G. Kim, T. P. Harrison, J. C. Hayya. 2004. The bullwhip effect – Impact of stochastic lead time, information quality, and information sharing: A simulation study. *Production and Operations Management* 13**:** 340-353.

Chen, F., R. Samroengraja. 2000. The stationary beer game. *Production and Operations Management* 9**:** 19-30.

Chickering, A., Z. Gamson, 1987. Seven principles for good practice. *AAHE Bulletin,* 39: 3–7.

Jacobs, F.R. 2000. Playing the beer distribution game over the internet. *Production and Operations Management* 9: 31-39.

Lee, H.L., V. Padmanabhan, S. Whang. 1997. The bullwhip effect in supply chains. *Sloan Management Review* 38: 93-102.

Prince, M. 2004. Does active learning work? A review of the research. *Journal of Engineering Education*, 93: 223-231.

Sterman, J.D. 1989. Modeling managerial behavior: Misperceptions of feedback in a dynamic decision-making experiment. *Management Science* 35: 321-339.

Warburton, R.D. 2004. An analytical investigation of the bullwhip effect. *Production and Operations Management* 13: 150-160.

AUTHOR BIOGRAPHIES

JEFF ADAMS is an Assistant Professor at the University of Indianapolis in the School of Business. He holds a B.A. from Miami University, an M.B.A. from Xavier University and a Ph.D. from George Washington University. His research interests include supply chain management, small business purchasing practices and logistics. His email address is jadams@uindy.edu.

JERRY FLATTO is an Associate Professor at the University of Indianapolis in the School of Business. He holds a B.S. in Mechanical Engineering from the University of Massachusetts, an M.S. in Computer Science and an MBA both from Rensselaer Polytechnic Institute and an Sc.D. in Management Systems from the University of New Haven. His research interests include valuing the worth of information systems and applying simulation techniques to supply chain management. His email address is jflatto@uindy.edu.

L. LESLIE GARDNER is a Professor at the University of Indianapolis with a joint appointment in the School of Business and the Department of Mathematics. She holds a B.A. degree in physics and mathematics from DePauw University, an M.S. degree in mathematics from Indiana State University, and M.S. and Ph.D. degrees in industrial engineering from Purdue University. Her research interests are in graph theory, particularly decomposition of 3-connected graphs and partial 3-trees. She is also involved in simulation modeling for optimization of profitability and costs in manufacturing and R&D environments. Her email address is lgardner@uindy.edu.