PRODUCTION AND OPERATIONS MANAGEMENT
Vol. 9, No. 1, Spring 2000
Printed in U.S.A.

EXPERIENCES WITH THE USE OF SUPPLY CHAIN MANAGEMENT SOFTWARE IN EDUCATION*

ANN CAMPBELL, JARROD GOENTZEL, AND MARTIN SAVELSBERGH
School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, Georgia 30332, *USA*

This paper discusses four experiments and experiences with the use of supply chain management software, in this case the CAPS Logistics software, at different levels of undergraduate and graduate education at the School of Industrial and Systems Engineering at the Georgia Institute of Technology. We hope that the readers will get an idea of the commitment and resources necessary to integrate supply chain software into the classroom as well as the potential rewards.
(LOGISTICS GAMES, SUPPLY CHAIN SOFTWARE, LOGISTICS EDUCATION)

1. Introduction

In the past decade, society has witnessed unprecedented changes in the availability and use of information technology. The use of e-mail and the Internet has become common place in industry as well as in many households. There has also been an increase in the availability of sophisticated planning tools, especially in the area of supply chain management, with vendors such as i2 Technologies, Manugistics, CAPS Logistics, and Numetrix.

Such changes should affect curricula at all educational levels in order to prepare students for the high tech world in which they will be living. Many people believe, for example, that it is essential that elementary school students are exposed to the Internet. Likewise, we feel, that it is necessary to expose undergraduate and graduate students in industrial engineering, management science, operations management, and operations research to sophisticated planning tools, such as supply chain management software.

This paper discusses four experiments and experiences with the use of supply chain management software, in this case the CAPS Logistics software, at different levels of undergraduate and graduate education at the School of Industrial and Systems Engineering at the Georgia Institute of Technology. We hope that the readers will get an idea of the commitment and resources necessary to integrate supply chain software into the classroom as well as the potential rewards. Our choice to use the CAPS Logistics software was one of convenience; it was available. However, there is no reason to believe that the experiments we discuss cannot be replicated with other supply chain management software.

In each of the experiments, students learn supply chain concepts in an environment that resembles one that they may encounter when they finish their education and go on into

* Received December 1998; revision received June 1999; accepted July 1999.

1059-1478/00/0901/066$1.25

industry. The use of supply chain management software makes teaching supply chain management more flexible and more extensive and provides students with more elaborate problem situations that better represent reality. Most supply chain software has well-developed graphical interfaces to represent the elements of the supply chain being studied. Visual representations help students connect better with the data and the solution methods.

We discuss the use of supply chain software in the form of an inventory routing game, as part of a case study and a design project, and as the basis for a software development class. Each of these situations represents a different level of interaction with the software. The inventory routing game can be used in a single lab session and does not require any previous exposure to supply chain management software. In a case study or a design project, where supply chain concepts are learned through model development, the use of supply chain management software is more involved and requires either previous experience or some form of basic training. Software development requires a much greater time commitment and has to be the focus of an entire class. In this case, students have to be or have to become intimately familiar with the supply chain management software and its capabilities.

The various situations also offer a continuum of interaction with reality. A game is internal to the class, removed from the source of data. A case study represents a company setting, perhaps incorporating some prepared data. In a design project, students work directly with company representatives and the data gathering process. In software development, students must draw from many experiences to anticipate operational reality.

The structure of the remainder of the paper is as follows. In Section 2, we briefly introduce the CAPS Logistics supply chain software. Section 3 describes our experiences with the use of a logistics game for vendor-managed inventory resupply. A supply chain design case study is presented in Section 4. The design project in Section 5 considers standard vehicle routing. Section 6 traces the software development of the inventory routing game used in Section 3. Finally, in Section 7, we summarize the advantages and disadvantages of using sophisticated planning tools in education based on our experiences. In order to facilitate replicating some of the experiments described in this paper, we provide more detailed information on each of them either in Appendices A–D or on The Logistics Institute web page at www.tli.gatech.edu.

2. Supply Chain Software

CAPS Logistics is a market leader in logistics modeling software. Its modeling capabilities are supplemented with a geographic information system (GIS) and optimization functionality. The core of all products is the Logistics Toolkit, which consists of logistically based data structures, modeling tools, and a macro-modeling language (MODL). Menu-driven platforms, created in the MODL language, provide specific modeling structures for a variety of routing and supply chain applications. The Logistics Toolkit allows further customization of each platform so that the models can be tailored to unique logistics scenarios. Logistics entities—such as plants, orders, distribution centers, customers, products, transportation channels, and vehicle routes—are made tangible through the graphical user interface (GUI) and up-to-date electronic maps and roads. CAPS software links with Microsoft Access via open database connectivity (ODBC).

Georgia Tech acquired the CAPS Logistics supply chain software when it joined CAPS Logistics Academic Link program. The Academic Link program offers CAPS Logistics software at no charge to its partners for use in teaching, research, and presentations. More information on this program is available at www.caps.com/partnerclients/academic/acadpart.cfm.

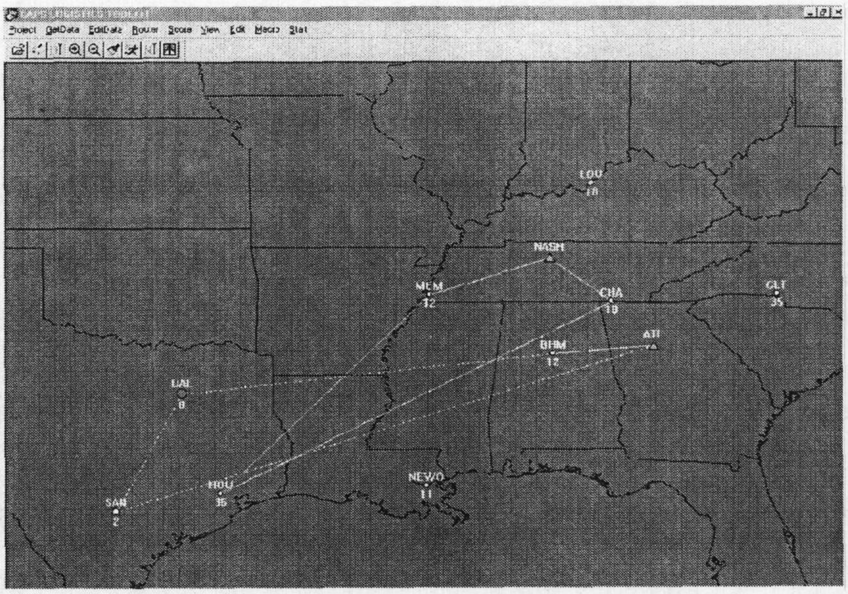

FIGURE 1. Screen capture of the logistics game.

3. Game Session

A simple way to introduce supply chain software in the classroom is by means of logistics games. Logistics games simulate the realities of a particular logistics problem, allowing the students (players) to develop a thorough understanding of the problem and to design and test various solution strategies in an engaging environment. Since the quality of the decisions made by the player are quantified by a score, students can compete with each other as in other games and make learning more fun.

There are several logistics games publicly available. In Appendix A, we provide a list of the ones that we are aware of and where they can be obtained. The main difference between the logistics game that we have used and the publicly available logistics games mentioned in the appendix is that the one we have used is built on top of commercial supply chain management software. This has the advantage that while students are playing the game and are learning about a specific logistics problem situation, they are also exposed to features of the underlying supply chain management software. Furthermore, the game has a professional user interface and extensive database capabilities, since they are "borrowed" from the supply chain management software (Figure 1).

We have used the logistics game in a master's level class on distribution systems. The class had just started studying routing and scheduling, and the logistics game considered vendor-managed inventory resupply. (Section 6 discusses the development of this logistics game in more detail.) The problem is to minimize the cost of supplying a set of customers over a certain planning period while trying to ensure that customers do not run out of product. When vendor managed resupply policies are in place, the vendor can make deliveries whenever he wants, and he can deliver any quantity that will fit in the customer's storage tank at the time of delivery. All customers use product at a customer-specific rate and have varying storage capabilities. Thus, a planner has to make decisions about who should receive a delivery each day, which vehicles and drivers should be used, what routes should be followed, and what quantities should be delivered.

In the game, the student assumes the role of planner and makes these decisions. The objective of the game is to minimize the total cost over the planning period. The total cost

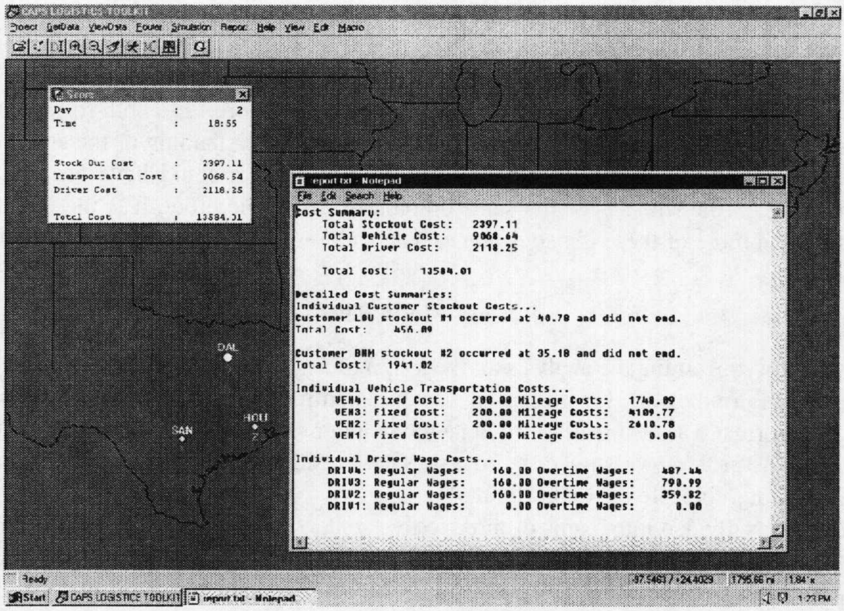

FIGURE 2. Report showing the score of a particular game play.

includes mileage costs, drivers' wages (both regular and overtime), fixed costs for using vehicles, and penalties for the length of time that a customer is out of product. At any time during the simulated planning period, the player (student) can construct new routes, provided that there are vehicles and drivers available at the distribution center. This involves using menus, clicking on customer icons, and entering delivery volumes, as prompted by information on the screen. These routes are executed, the appropriate costs are incurred, the internal database is updated, and the graphical representation of the state of the system changes accordingly. During the planning session it is possible to look at helpful statistics about the customers and their inventories, the vehicles, as well as at information about what has been planned and not yet executed. All of this is an effort to make the game as close to reality as possible so that the students can understand and appreciate the complexity and difficulty of the problem, but in as simple and entertaining a way as possible (Figure 2).

We installed the inventory routing game on several computers in a laboratory setting. Students were divided into groups, and a sample "play" of the game was demonstrated to all of them. They were given 30 minutes to get familiar with the game before playing the game through the entire planning period trying to get the best score possible. While getting familiar with the game, students quickly got a good understanding of the complexities associated with implementing a vendor managed inventory resupply policy by experiencing the impact on cost of different decisions. The graphics helped the students form ideas for routing strategies. From experimenting, they also quickly realized the benefits of planning ahead and considering the long term rather than just the short term, especially when resources are tight. For example, they might plan on sending a truck from Atlanta to Dallas without considering making deliveries to customers along the way earlier than necessary. They often realize a little later, when this truck is on its way to Dallas and all of the vehicles are tied up, that the customers in between are getting close to running out. Experimentation also taught the students what information was important to track for the customers.

Logistics games provide a simple way to help students gain a thorough understanding of complex logistics problems. They experience very directly what makes these problems difficult. By experimenting, they form ideas both on how to and how not to solve them. As mentioned earlier, in our case, they were also exposed, although in a limited way, to supply chain management software.

The overall experience was very positive, even though the students felt that they would have been able to do better and learn more if they had had more time both to practice and to play the game. However, it was clear that the students' understanding of the complexities of vendor-managed inventory resupply was much greater than could have been expected from lecturing on the subject for the same amount of time. Therefore, it is the intention to develop several more of these logistics games, each addressing a different logistics situation.

4. Case Study Class

A case study concerning the analysis and redesign of a North American distribution system was used as the framework for a focused supply chain modeling course for undergraduate seniors. The format was designed to simulate the work environment that students would soon experience. The professor assumed the role of Vice President for Logistics and the teaching assistant assumed the role of a consultant.

The class was divided into teams of three to five students. All teams were "in training" for the company, temporarily assigned to the VP for Logistics for the length of a school term. Two of the three class projects were designed to use the supply chain software. For that reason, software training occurred in the second week of the 10-week term. The teams remained the same throughout the term. Lectures were provided only when issues arose where the teams required additional education. Teams worked with the software in their own time. No official lab session was established.

The case was based on the supply chain used to deliver service and repair parts to dealers for Ford Motor Company. Drawing upon an existing research relationship with the planning department at Ford, extensive data were available for the case and several decision models had already been developed. Since the term was limited to 10 weeks, it was decided to begin with an established model rather than have students develop their own. The consultant proposed the model, a simplified network flow formulation of the Ford supply chain already implemented in the CAPS software. Students were encouraged to challenge the validity of the model to gain an understanding of the relevant supply chain issues and then to improve and modify it to develop modeling skills. Figure 3 shows the sites and arcs for the network model. In addition, students were furnished with spreadsheets of the data used to establish model cost, demand, and capacity parameters.

The initial project was the same for all groups: determine the number and location of regional distribution centers. The students very much enjoyed the ability to quickly evaluate the effect of moving distribution center locations, both in terms of overall cost and channel decision visualization. Such rapid replication and visualization would not have been possible with a standard paper case.

The second project assigned a different objective to each group, though each was related to the modeling of distribution center to dealer transportation. The network model utilized less-than-truckload (LTL) carrier costs, though most of the dealers received product via multistop routes. One group contemplated how to better consider routes in the model. The second group considered the effect of delivery frequency. The third group determined the merit of using pool points instead of LTL carriers.

Ideally, the initial project would have familiarized students with the software, allowing them to modify the proposed model as needed for the second project. However, the students had not grappled enough with the model in the first project. As a result, they were not able to fully utilize the software capabilities in the second project.

It became clear that the initial project should have been more narrowly defined, allowing the students to implement some simple ideas and concepts and to compare these to the default options provided by the software. In this way, the students would have gained confidence in

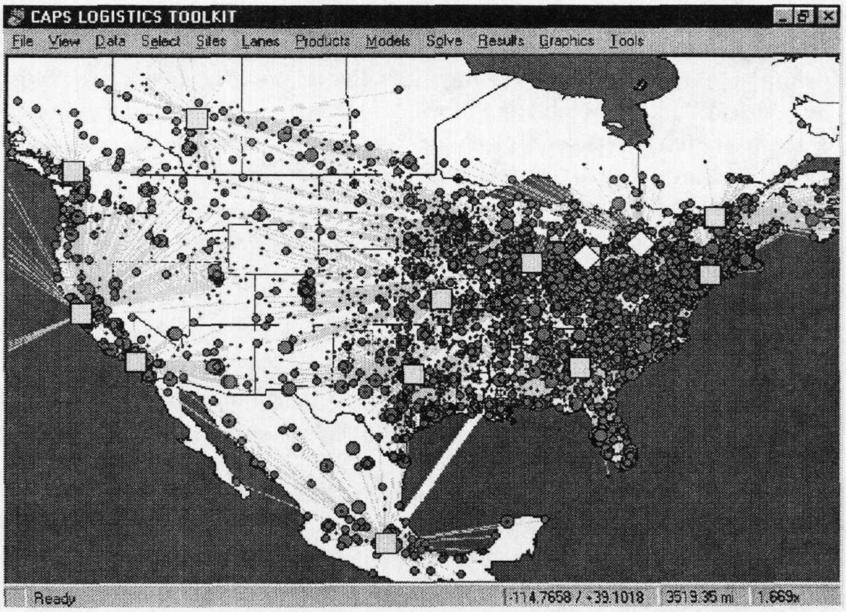

FIGURE 3. Network flow formulation for the supply chain modeling case. Diamonds represent the replenishment depot for Canada and U.S./Mexico. □, regional distribution centers; ○, dealers, scaled according to dollar demand.

their software application skills. Course evaluations resonated with this idea. One pointedly said that the class should have started with a "case that was simple enough to have a correct 'answer' so that students would get feedback as to whether they were using (the software) correctly or not."

When students are supposed to adapt and/or develop a model for advanced case analysis, then they need to begin by constructing the model—in their words, "to start from scratch." Though more time consuming, the students would acquire software modeling skills and become more versed in the technology. This interaction with the software should take place in a defined lab session where a "lab advisor," with advanced knowledge of the software, is available for consultation.

Despite struggles with the second project, students warmly welcomed the integration of software with the case study. One student wrote, "The [software] exposure was very beneficial. We were introduced to a real world technology and how to apply it to a real world problem." Software enabled a case study with a mixture of rich data and advanced modeling technology from which the students gained insight on how to approach supply chain problems.

5. Design Project

A design project—or independent study—affords the opportunity to immerse the student in supply chain software. With more focus and a longer time scale than a case, a project allows enough time for effective self-paced learning. Students reap the benefit of this training by evaluating many more solutions and scenarios than would be possible without software.

A design project is part of the core curriculum for industrial engineering undergraduates at Georgia Tech. Titled Senior Design, this project-oriented class is taken during the student's final two quarters. The students are organized into groups of no more than five. Projects are selected from a list of company-submitted proposals, which describe a problem and designate a company representative. One faculty member is assigned as advisor for the group, matching expertise with problem definition.

To illustrate the integration of supply chain software with a Senior Design project, we follow one group through the 6-month process. This particular group of five students chose to work with the Atlanta region of the American Red Cross. The Red Cross suggested to investigate the distribution of blood and blood components to 115 hospitals throughout the state of Georgia. Currently, hospitals place orders daily. Deliveries are either made with Red Cross vehicles that are manually routed each day or sent directly via a courier service or taxicab. Red Cross sought a framework to guide the manual process. Moreover, they aimed to lessen reliance on the expensive courier service. The project targeted these two objectives.

After consultation with the faculty advisor and company representative, data requirements were outlined, and solution approaches were proposed. Route data sheets used by the dispatcher were collected for a 2-week period, capturing order quantities, load/unload time, driving time, and distance between stops. Vehicle, driver, and courier cost parameters were collected to value solutions for this fortnight scenario. Current routes establish a baseline for assessing solution approaches. Space-filling curve (SPC) and nearest neighbor route generation heuristics were proposed.

The group initially planned to evaluate solutions manually using trip mileage technology (at the website www.mapquest.com) and a matrix lookup. The faculty advisor suggested CAPS Logistics software for both route generation and route evaluation.

By the end of the first quarter the group had invoked the software's geographic information system (GIS) functionality to geocode hospitals—translating the address into latitude–longitude. The geographic representation provided a visual understanding of the delivery problem and vector coordinates for the heuristic algorithms. This hour-long process would have been onerous without the software.

At the beginning of the second quarter, students began entering the 2-week order scenario. The need to define orders and vehicles for the software focused the modeling effort. The students had to understand hospital order timing, distinguishing AM and PM orders. Forced to match orders with vehicle characteristics, they determined that vehicle capacity was not a limiting factor since Red Cross vehicles can handle significant peaks in order quantities. Instead, route length in time was the crucial constraint. Software functionality also prompted consideration of time windows. Students determined that although time windows may be important for operation, they were not required for developing the planning framework. Though these issues may have arisen anyway, the software provided a natural setting to guide students through the key modeling issues.

The final report recommended a mixture of the two heuristics as a framework for manual routing. The SPC heuristic was used to route two vehicles for AM deliveries in the Atlanta metropolitan area. They determined that SPC is appropriate where the road network is dense. The nearest neighbor heuristic was used to set the fixed route basis for PM deliveries throughout the entire region (see Figure 4). Eight trucks are used for these routes. Based on results for 2 weeks of orders, the annual transportation cost was reduced by nearly 9%, from \$631,000 to \$577,000. Routing vehicles over an electronic road network provided credibility for Red Cross managers. Moreover, the software allowed swift, detailed examination of multiple days' solutions. Creating the same visualization and credibility would have been a monumental task without the use of software.

During job interviews, students working on this project have found companies to be very interested in this software experience. In addition to logistics insight, students developed GIS skills and understand of the geocoding process and other issues related to using electronic road networks.

It was observed during the course of the project that students needed a fair amount of time to experiment and familiarize themselves with the supply chain software functionality. Translating raw data into the right form for the software takes time, several weeks in this case. The benefit is that once data are formed appropriately, solution adaptation and replication is speedy. As with the controlled game environment, interaction with many

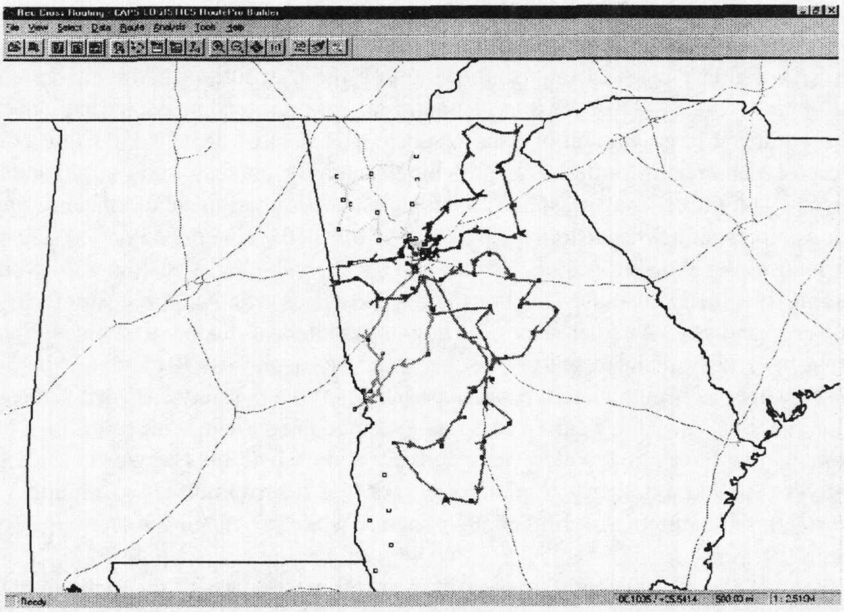

FIGURE 4. Fixed route structure for Atlanta Red Cross deliveries.

solutions develops insight rapidly. Since this final phase is where much of the supply chain intuition is developed, keeping to an aggressive schedule is worthwhile.

6. Software Development Class

The final example of the use of supply chain software in the classroom is a 10-week course devoted to the development of a logistics game. The objective in the course was to design and implement a logistics game for a complex logistics problem. The game had to have a graphical interface that would visualize the state of the system as it evolves over time and had to allow a decision maker (the player) to influence the behavior of the system. To achieve this, the class combined a simulation with the visualization and database capabilities of the supply chain software. During the course, students learned more about the specific logistics problem, logistics software, discrete event simulation, interface design, and software engineering. The level of understanding of the supply chain software necessary for this course goes well beyond that required for the earlier examples.

The logistics game developed is the inventory routing game mentioned in Section 3. The group started by developing a thorough understanding of vendor managed inventory replenishment systems by identifying the key objects and relationships involved in such a system. It is essential to understand the relation between usage patterns, vehicle capacities, tank capacities, routing decisions, travel times, etc., in order to create an effective and realistic simulation. Creating state transition diagrams that describe every possible interaction was time consuming, but it was important in obtaining an understanding of how all the factors were related. While doing this, the group identified what data are needed to initialize the simulation, what data are needed to describe the "state" of the system, and what data would need to be maintained. With this in mind, they created an appropriate relational database structure.

After designing the basic interactions to be modeled in the game and creating the database, the class was divided into two groups. One group was assigned the task of developing the simulation (done in C++). The main task of the simulation is to generate the next event based on the current state of the system and the current set of specified routes. The second group was charged with using the supply chain software to create the user interface and to maintain and update the data representing the state of the system.

One of the primary tasks of the group designing the user interface was to represent the "current state of the system." This is a nontrivial task in a complex system, because it typically requires a large amount of data to specify the state of the system. Choices include how to convey as much information as possible through the choice of colors, shape of icons, size of icons, and limited statistics. For example, what information should the size and color of the icons representing the customers give? In the finished game, the color and size indicate whether a customer is using product (small and green), whether a customer's inventory is below some specified threshold (medium and yellow), or whether the customer is out of product (large and red). Another issue was how to handle and display the routes. Should all routes that have been planned but not yet executed be displayed? If they are, the user can quickly see what has been planned, but it may also lead to a possibly crowded screen. In the finished game, only the routes currently being executed are shown, with solid links between the customers that have already received a delivery and dashed links between customers that have not yet received a delivery. Another key task was determining how to guide a novice player through the game by offering enough "help," while not making it annoying for more experienced players.

By the end of the quarter, both groups had their respective parts of the game in place, but there were still some outstanding issues regarding the information passing between the two parts that needed to be worked out. One class member from each group worked on the project during the next quarter to get the game fully functional.

Maybe even more so than in the previous examples, it took the students a long time to get to the point where they felt comfortable enough with the software to carry out the required tasks. Members of the class received basic training on the software early, but did not really start using it until the design of the game was completed. A better understanding of the software early on would have helped in the design of the simulation. It may also have been better to create the teams early on, so that they could have started discussing implementation choices and done some initial coding before the design of the game was complete, increasing the chances of having a working prototype by the end of the quarter.

Using the supply chain software helped speed up the development greatly. Furthermore, being able to call on existing "tools," especially for graphics, enabled the creation of a game that looks professional. The built-in macro language made designing database updates that translate into changing icons and colors on the screen relatively simple. The internal database structure of the supply chain software was easy to use and made maintaining and checking data values fairly straightforward.

Ideally, the inventory routing game would be made available as an executable for students (and others interested) to use at home. Unfortunately, the CAPS Logistics Toolkit does not support the creation of such an executable at the moment. Therefore, the use of the game is currently limited to computers with a license to the CAPS supply chain management software.

7. Conclusions

Overall, our experiences with the use of supply chain software in undergraduate and graduate education have been positive. We conclude by summarizing the advantages and disadvantages of using sophisticated planning software in the class room as we see them. First and foremost, students seem to like it. As a consequence, students participate more actively and absorb more of the material presented to them. Second, the use of sophisticated software allows the instructor to illustrate the material using realistic size instances that reflect the true complexity of the planning problems. Furthermore, using planning software makes it easy to experiment with various instances as well as various solution strategies, all of which lead to a more thorough understanding of the planning problem at hand. Third, most sophisticated planning tools have graphical user interfaces that visualize instances and solutions. Such visualizations are informative and facilitate understanding specific characteristics of the problems. (A picture is worth a thousand words.) Finally, students are exposed to the type of sophisticated software that they may encounter when they finish their education and go into industry. In fact, it increases their chances of finding a job, since many of the companies appreciate experience with sophisticated planning tools.

On the other hand, there are also some disadvantages to the use of sophisticated planning tools. First, as was the case with the supply chain software used in the experiments described in this paper, we are typically dealing with commercial software. This means that the use of the software may not be free of charge, that the use of the software may be restricted, that there may be complicated licensing issues, or that the vendor is only prepared to provide support and training at a cost. Second, the use of software requires appropriate computing facilities and computer support staff. Third, in most cases the use of software, especially sophisticated software, requires a large upfront time investment on the part of the instructor. Many academic institutions do not have a reward structure, in terms of tenure and promotion, that entices faculty to undertake such efforts. Finally, courses involving the use of sophisticated planning software may need to be structured differently. The single biggest complaint from students was that there was not enough time to learn how to use the software. This is not surprising given the complexity of sophisticated planning tools. It would be nice if students could familiarize themselves with the software at home, but this is usually not possible since it is commercial software.

Following is a set of appendices providing more information about other logistics games (Appendix A), our procedure for developing a game using supply chain software (Appendix B), and descriptions of the format of the case studies we have made available (Appendices C and D). The data sets for the case studies are Microsoft Excel documents and do not require CAPS or other supply chain management software to view them or to use them.[1]

[1] We thank the people at CAPS Logistics for the support that they have provided during the activities discussed in this paper.

Appendix A

Logistics Games

The most famous logistic game is probably the Beer Distribution game developed by John Sterman at MIT (Sterman 1989, 1992) (learning.mit.edu/pra/tool/beer.html). It can be played using paper and a few supplies or online using the version implemented at the University of Indiana (Jacobs 2000) (jacobs.indiana.edu/beer). The game involves four entities: a retailer, a wholesaler, a distributor, and a factory. The objective is for each player (entity) to minimize costs, where costs are based on inventory carrying costs and backlog charges.

Jackson and Muckstadt at Cornell University have developed several simulation games, including a distribution game (www.orie.cornell.edu/~jackson/distgame.html), a transportation game (www.orie.cornell.edu/~jackson/trucks.html), and a warehouse location game (www.orie.cornell.edu/~jackson/whsloc.html). The games in this suite are easy to understand, take little time to get started, and are nice graphically. The distribution game involves both a supplier and a central warehouse and requires the user to make supply decisions to meet random demands at multiple locations. The transportation game is concerned more with learning how to make effective routing and scheduling decisions for trucks given a set of demands. The warehouse location program allows the user to locate warehouses, as well as plan truck routes. Also available on Jackson's website is a sampler of multimedia virtual tour of a factory (Jackson and Muckstadt 1990) (www.orie.cornell.edu/~jackson/plnttour.html).

Robert Grubbstrom from the Linköping Institute of Technology in Sweden has developed the international logistics management game (a demo is available at www.ilmg.com). It involves up to seven competing companies, production in four regions, multiple modes of transportation, varying wage and production levels in different markets, and allows communication among the players (companies).

Lean Production is an interactive simulation game modeling a bicycle factory and its supply chain. The game includes a logistics planning and control system based on the MRP II concept and an integrated controlling information system including business planning and performance indicator systems (Zapfel and Piekarz 2000) (www.ifw.unilinz.ac.at/lean.html).

The number of computerized logistics games appears to be increasing, since two more games "in-progress" were found in a web search. The Michigan Interactive Logistics Simulation game developed by Dennis Severance and David Murray is in the process of being made playable online (mis.huji.ac.il/Mils). The Canadian Professional Logistics Institute is producing an interactive CD that simulates a logistics problem considering seasons, shelf-life, and international suppliers and customers. It is not available yet, but the samples viewable on the web at www.loginstitute.ca/cdrom.html look impressive.

Appendix B

Developing a Logistics Game

In this appendix, we provide more details about the development of the inventory routing game, for those interested in creating logistic games themselves.

At the heart of the game is a relational database that stores all the information necessary to describe the state of the system, the planned routes, and the executed routes. The database is initialized with the data describing a specific problem instance. To facilitate playing different instances, an Access database was designed to store all the data necessary to describe an instance. Given a sample Access database, anyone should be able to easily create an instance that represents a variation of the problem in which they are interested. This instance can then immediately be used in the game. In the game, the user selects which Access database to import and the data are imported into the CAPS internal relational database via ODBC. As soon as the data are present in the CAPS database, the user will see a visual representation of the initial state of the system via the graphical interface. The CAPS internal database stores a lot more information than the initial Access database, since the game needs to keep track of the changes that will occur during the planning period as well as the planned and executed routes. This includes, for example, whether each customer is using product or not and where each vehicle is heading from and to.

While playing the game, part of the information stored in the CAPS database is represented graphically on the screen, such as customer locations and partial information concerning their status (above safety stock, below safety stock, or out of product). In addition, the user can use the on screen menus to query the CAPS database and get more detailed information on the customers, vehicles, products, and drivers.

At the start of the game, the user must decide which events (vehicle departs from plant, vehicle arrives at customer, customer starts using product, customers hits safety stock, etc.) will force the simulation/game to pause, so the user can see and evaluate the new state of the system and, if the need arises, can plan new routes. After these decisions are made, the simulation is initiated. This means that all the information in the CAPS database is written out to text files. These text files are used by the simulation to create an initial event queue. The simulation is written in C++ and converted to a dynamic link library (DLL). The game itself is written in MODL, the macro language provided with the CAPS logistics toolkit. At appropriate points in time, the game calls the simulation functions. As in a standard discrete time event simulation, the event queue contains all of the known upcoming events for each customer, ordered by their estimated time of occurrence. When the simulation is called it advances its internal clock until the first event occurs that is of one of the types specified by the user as a stopping event. At this point, the simulation hands back control to the game and passes back an array of values describing what event has just happened and at what time the event took place. Using this information, the game updates all of the time-dependent information in the CAPS database, such as anything involving usage, to give it the appropriate value at the current time. Other information in the array includes which customer, vehicle, and driver (if any) were involved in the most recent event and relevant volumes. This allows CAPS to determine which parts of the planned routes have now been completed and how and when this happened. After the CAPS database has been updated as a result of the event information, the screen is redrawn to reflect the new state of the system. Also included in the array are the values representing the three parts of the score: the driver cost, reflecting hourly wage and overtime costs; the vehicle cost, reflecting a fixed cost for usage and a per mile charge; and stockout costs, reflecting a per unit charge for any customers having deficient inventory. After the database is updated and the screen redrawn, the user has the ability to use the menus to look up information or create any new routes before selecting *resume*. After selecting resume, certain parts of the CAPS database are rewritten to text files, and the control is handed over to the simulation. To maintain the event queue, the DLL for the simulation is not freed until a complete play of the game is complete. Effectively, the simulation just resumes after every event beyond the initial one. The game proceeds in this same way with control passing back and forth until the user decides to end the game or the playing horizon (specified in the original Access database) is reached.

From the above discussion it is clear that we have relied heaviliy on two features of the CAPS Logistics Toolkit: the internal database and the graphics capabilities (including the use of menus, etc.). To reproduce this game or to create one similar to it, institutions that do not have access to the CAPS Logistics Toolkit need access to other supply chain software that can provide similar functionality and features, or need to implement these themselves in standard programming languages such as JAVA. However, the use of a standard programming language will increase the development time significantly.

Appendix C

Inventory Routing Problem Instances

The IRP is concerned with the repeated distribution of a set of products from several facilities to a set of customers over a given planning horizon. The facilities can produce these products at given rates and have ample storage capabilities for the products. The customers consume products at a given rate and have limited storage capabilities. A fleet of vehicles is available at each of the facilities as well as a set of drivers. The objective is to minimize the overall costs during the planning period.

The following data have been collected for two real-life instances of the IRP and are available on The Logistics Institute web page at www.tli.gatech.edu/research/casestudy/cs.htm in the form of several Excel workbooks.

Customers

- ID: an identifier for each customer—such as a business name or city where it is located.
- X: longitude coordinate.
- Y: latitude coordinate.
- OPENTIME: time a customer starts using product each day (using a 24-hour clock, like all times).
- CLOSETIME: time a customer stops using product each day.
- OPENWINDOW: time a customer starts being able to receive deliveries each day.
- CLOSEWINDOW: time a customer stops being able to receive deliveries each day.
- FIXEDSTOP: fraction of an hour required to make a stop at a customer, not including fill time.
- MATEVEHCLASS: type of vehicle able to make delivery at a customer.
- PRODTYPE: type of product that is used by a customer (limit of one currently).
- PRODMEAN: mean rate at which customer uses product per hour when time is between open time and close time.
- PRODSTDEV: standard deviation of this usage rate (not used currently).
- PRODCAPACITY: the limit on how much inventory of a product can be held at a customer.
- PRODSS: "safety stock" for inventory of a product. It is often set so that when inventory falls below this level, this is a trigger to plan a delivery to the customer.
- PRODINV: initial inventory of product at a customer.
- PRODSOCOST: when customer runs out of product, this is the "cost" per unit that the customer would have used if sufficient resources were available.

Drivers

- DRIVERID: an identifier for each driver.
- HOMEBASE: home facility associated with a driver.
- OPENWINDOW: time a driver can start driving each day (not enforced).
- CLOSEWINDOW: time a driver must return to the home plant each day (not enforced).
- REGTIMEWAGE: wages earned per hour of regular time work.
- OVERTIMEWAGE: wages earned per hour of overtime work.
- MATEVEHDRIV: type of vehicle that a driver is able to drive (not enforced).

Facilities

- ID: an identifier for each plant such as the city where it is located.
- X: longitude coordinate.
- Y: latitude coordinate.
- OPENTIME: time plant starts producing (the same for all products currently).
- CLOSETIME: time plant stops producing (the same for all products currently).
- FIXEDSTOP: fraction of an hour required to make a stop at a plant while driver is on a tour, not including reload time.
- FAILUREPARM: parameter for describing frequency of failure in production process (not used).
- MATEVEHCLASS: type of vehicle that is able to pick up product from a plant (not used).

Products

- PRODUCTID: an identifier for each product.
- FILLRATE: number of units of product per hour that can be pumped into a vehicle at the plant.
- DISPENSERATE: number of units of product per hour that can be dispensed from a vehicle to a customer.

Facproducts

- ID: unique record number representing a plant/product pair.
- FACILITYID: an identifier for a plant, must match an ID on FACILITIES worksheet.
- PRODUCTID: an identifier for a product, must match a ProductID on PRODUCTS worksheet.
- PRODRATE: number of units of the product that is produced per hour while plant is producing.

- PRODINV: initial inventory at a plant of a product.
- PRODCAP: limit on amount of inventory of a product that can be maintained at a plant.

Vehicles

- VEHICLEID: identifier for each vehicle.
- HOMEBASE: home facility associated with a vehicle.
- MAXVOLUME: limit on amount of product that a vehicle can hold.
- SPEED: speed at which vehicle drives on average, used to compute travel times.
- FIXEDCOST: cost charged for using a vehicle during the time horizon.
- COSTMILE: cost charged per mile driven on a vehicle.
- PRODUCTID: an identifier for a product, must match a ProductID on PRODUCTS worksheet.
- MATEVEHDRIV: "type" of vehicle (not used).
- VEHFAILPARM: parameter for describing frequency of failure in delivery process (not used).

Instance

- INSTANCEID: identifier for instance.
- TIMEHORIZON: number of days game will be played.
- DRIVLIMIT: driving below this limit will be charged regular wage, above this will be charged overtime wage.
- USAGECHINT: (not used)
- STOPOPEN: (not used)
- STOPCLOSE: (not used)

The following basic questions related to the inventory routing can be investigated:

1. How would you decide which customers should receive a delivery on a day to make sure none of them would run out of product? Would you just look at current inventory or would you look at distance from the plant as well?
2. Which customers do you think would be good choices to be on a route together? What factors would you use to make such a decision?
3. If you were a planner trying to make a schedule for these customers, is there any other information that you think would be helpful?
4. If you were making a schedule for delivering to these customers, how far do you think you would plan ahead to make sure you wouldn't let anyone run out of product? 1 day? 2 days? Why?
5. For a given dataset, which appears to drive the total cost more: stockout cost, driver costs, or vehicle costs? For each of these, if it represented the only cost involved, how would this change your delivery policy?
6. Time horizon is listed as a characteristic of the instance. How do you think strategy would be different if time horizon was 3 days versus 33 days? Why?
7. How do small delivery time windows for customers complicate the problem? Do small windows for using product really affect things?
8. Do you think it would make the problem easier or harder if all customers had product capacity the same size as vehicle capacity? Why?
9. If you consider the stochastic information about customer usage rate, such as customer specific standard deviation, would this change your answer to question 1, and if so, how?
10. If safety stock is used as a signal to start planning a delivery to a given customer, how would you suggest setting this level? What factors would you consider besides usage rate?

Appendix D

Supply Chain Design Case Study

The supply chain design case study focuses on the distribution of automotive parts and supplies to the Ford authorized dealers throughout North America. Ford is faced with pressure to provide excellent customer service, which means timely distribution of parts to the dealers, with minimal logistics investment, both in capital and operations. The design of supply chain infrastructure will have a strategic impact on this objective.

Ford authorized parts flow through supply chain infrastructure that has been used for years and consists of a National Replenishment Center (RC), several Regional Distribution Centers (DCs), and Dealers. The following data have been collected to assist in the evaluation of the existing supply chain and in the construction and analysis of alternative supply chains:

Dealers

- Location: latitude, longitude, and zip code
- Current primary DC
- Current route assignment
- Demand
- Average shipment size

Regional Distribution Centers

- Location: latitude, longitude, and zip code
- Material handling cost
- Fixed cost
- Inventory cost

National Distribution Center

- Location: latitude, longitude, and zip code

Products

- Weight
- Value

Lanes

- Distance
- DC replenishment time, including handling and transportation
- Origin-destination cost quotes for truck and rail
- Carrier contract rates for multistop routes
- LTL cost (nondiscounted) for average shipment size

These data, as well as a more thorough description of them, are available on The Logistics Institute web page at www.tli.gatech.edu/research/casestudy/cs.htm in the form of several Excel workbooks.

The following basic questions are related to the supply chain:

1. Does the current supply chain have the right number of distribution centers and are they placed in the correct locations?

2. Consultants have performed an analysis assuming dealers are allocated to the nearest distribution center. Does such an allocation result in minimal supply chain costs? If not, how should the dealers be allocated to distribution centers?

Other issues that may be addressed are as follows:

1. Dealers are typically visited on multistop routes from a distribution center. There are basically two approaches to modeling multistop deliveries in a network flow structure (the basis for most supply chain design models): (1) fixed dealer clusters, and (2) individual dealers, where the "route costs" are suitably divided over individual dealers. How does the chosen approach affect the resulting supply chain? How should the "routes cost" be divided over individual dealers?

2. Ford makes regular visits to the dealers. It is currently assumed that each dealer is visited three times a week. This allows the computation of average shipment sizes by calculating the overall demand and dividing by the delivery frequency. Average shipment sizes have been used in the supply chain design process. Is this realistic? How does it affect the resulting supply chain?

3. Ford is considering changing to every day delivery. How is this going to affect the costs?

4. Ford is considering the use of pool points. Pool points are locations that provide a similar function as a distribution center, namely transhipment, but without any storage facilities. Pool points can be thought of as parking lots where trailers or loads can be exchanged. Is this a viable option? Where would you locate pool points? How would you operate a supply chain including pool points?

5. Up to now we have not discussed issues related to inventory costs at the distribution centers. How can these be incorporated in the models?

References

JACKSON, P. AND MUCKSTADT, J. (1990), *Llenroc Plastics: A Case Study in Manufacturing and Distribution Systems Integration*, Technical Report 898, Cornell University Department of Operations Research and Industrial Engineering.

JACOBS, B. (2000), "Playing the Beer Distribution Game Over the Internet," *Production and Operations Management*, 9, 1, 31–37.

STERMAN, J. (1989), "Modeling Managerial Behavior: Misperceptions of Feedback in a Dynamic Decision Making Experiment," *Management Science*, 35, 321–339.

STERMAN, J. (1992), "Teaching Takes Off: Flight Simulators for Management Education," *OR/MS Today*, 40–43.

ZAPFEL, G. AND PIEKARZ, B. (2000), "The PC-Based Simulation Game Lean Production for Controlling the Supply Chain of a Virtual Bicycle Factory," *Teaching Supply Chain Management*, Vol. 2 of *POMS Series in Technology and Operations*.

Ann Campbell is a Ph.D. candidate in the School of Industrial and Systems Engineering at the Georgia Institute of Technology. Her research focuses on integrating inventory and routing decisions,

and is motivated, in part, by her work with Praxair, an international industrial gases company that has adopted a vendor managed inventory replenishment policy.

Jarrod Goentzel received his Ph.D. from the School of Industrial and Systems Engineering at the Georgia Institute of Technology. His research focused on supply chain design and was motivated, in part, by his work with Ford Customer Service parts distribution in North America. He is currently Director of Product Management for supply chain products at Baan Company.

Martin Savelsbergh is Associate Professor in the School of Industrial and Systems Engineering at the Georgia Institute of Technology. His research focuses on computational integer programming and computational logistics. His current work includes projects in mixed integer optimization, vendor managed inventory replenishment, and dynamic and stochastic vehicle routing.