

Writing MIS Mini-Cases To Enhance Cooperative Learning: A Theory Of Constraints Approach

Danilo Sirias

Department of Management and Marketing
Saginaw Valley State University
University Center, MI 48710, USA
dsirias@svsu.edu

ABSTRACT

Teaching Introduction to Management Information Systems (MIS) courses is a formidable challenge because such teaching entails covering a relatively large, ever-changing subject, as well as finding the right balance between audiences with different expectations and levels of knowledge. The literature suggests cooperative learning as a viable strategy to teach MIS effectively. In a cooperative learning environment, students can bring their expertise to the table and work together as teams to solve business problems. One strategy to support cooperative learning in an MIS class is teaching through case studies. Since cases tend to be too long to be used during a regular class session, this paper proposes the use of mini-cases to enhance cooperative learning. The paper also provides a procedure to write mini-cases based on one of the thinking processes within the Theory of Constraints.

Keywords: Cooperative learning, Theory of Constraints, Cloud, Mini-cases, MIS

1. INTRODUCTION

Teaching Introduction to Management Information Systems (MIS) courses is a formidable challenge. First, MIS draws research and principles from different disciplines (i.e., psychology, economics, and sociology). As a result, instructors are expected to understand a large variety of subjects and be able to present them in a meaningful and interrelated sequence.

Due to the variety of topics, students may tend to become confused and lose focus. To add to the level of complexity, MIS touches every function of an organization (Jacobs and Whybark 2000). A traditional Introduction to MIS class covers aspects in disciplines such as human resources, accounting, marketing, and financial information systems. On top of all this, MIS is an area moving at lightning speed, making class preparation a difficult task.

Yet another problem could come from the fact that students in an introductory MIS course may have different educational backgrounds (computer science, business, etc.). In some instances, Computer Science

departments may require their students to take at least one MIS class, to make sure they understand the role they are going to play in the business world. However, computer science students' expectations as to what MIS entails differ considerably from those of business students. Computer science students usually expect some technological focus and hands-on type activities similar to typical computer science courses. Business students also expect to learn something about technology, but not at the same depth. They also want to see how everything fits within a business environment and how technology can help businesses.

Another situation coming from having dissimilar types of students is different levels of knowledge about computers. A class may have students with a high level of programming skills in different languages as well as students who are still afraid they may break a computer if they touch it. Professors must therefore decide the depth to which technology should be covered. In short, the challenge of teaching MIS entails covering a relatively large, ever-changing subject, as well as finding the right balance between audiences with

different expectations and levels of knowledge.

2. COOPERATIVE LEARNING MODEL TO TEACH MIS

Having different levels of computer knowledge as well as different expectations can become an advantage if we exploit the various levels of expertise and combine them into a learning experience, one of the premises of working in teams within a business environment, but the same principle can apply to learning by allowing students to bring their individual perspectives into the class. Group learning exercises for MIS have been recommended in the academic literature (Fellers 1996). The theoretical background underlying learning in teams is captured under the cooperative learning model, in which students learn through “interaction of individuals with other individuals” (Leidner and Jarvenpaa 1995, p. 268). In an MIS class, students can bring their expertise to the table and work together as teams to solve business problems.

The literature covering the advantages of cooperative learning is vast. A meta-analysis of over 300 studies favorably compared cooperative learning with other learning approaches (Johnson, et al. 1998). Slavin (1990) suggests that better understanding comes from students interacting with each other. Whipple (1987) argues that ideas coming from different points of view can result in new, shared knowledge. Cooperative learning has been associated with improving creativity, generating better ideas, developing critical thinking, as well as fostering higher content retention (Schlechter 1990). Cooperative learning is not only confined to teaching in the classroom but has also been used for training in companies (Newstrom and Lengnick-Hall 1991).

One strategy to use cooperative learning in a management information systems class is teaching through case studies (Granger and Lippert 1999). Cases are widely used as a teaching method for business courses because they provide an opportunity for students to simulate “real” world situations and also to improve their problem-solving skills. Cases can also be designed so that managerial problems can be attacked from different perspectives, including a technical one. Working on cases in teams mirrors the reality of the workplace and enhances practical skills.

However, regular case studies may be too large to be used as part of a regular class session. To solve this situation, mini-cases (which can potentially be solved via a predetermined technology, e.g., data mining, virtual private networks; maximum one page long) can be used; these are very focused and have solutions related to relevant MIS concepts. Analyzing mini-cases

has many advantages. First, it provides relevance to the material being studied. Furthermore, students gain knowledge about applications of MIS to different business situations and learn the impact technology can have in businesses (Mukherjee and Cox 2001). The idea is to have mini-cases that can be worked in interdisciplinary teams of three or four students; such teams are desirable to provide an ideal environment for participation within the group (Leidner and Jarvenpaa 1995).

The problem is that most textbooks are not written with such instructional methodology in mind. Some textbooks may offer mini-cases, but they may not be targeted to a specific technology. Also, very often mini-cases in current textbooks merely describe what a company already did, leaving no room for students to devise a solution themselves. Hence, there is a need to develop a procedure which will aid instructors to write good mini-cases that can be used by interdisciplinary teams in a class setting.

The question is: what elements should be included in a mini case so that it provides sufficient information to be challenging for students? The procedure presented in this paper uses a model based on one of the thinking processes of the Theory of Constraints (TOC). The paper presents an overview of the Theory of Constraints and a description of one of the thinking tools of TOC, referred to as the cloud. Then, it describes a procedure to write mini-cases based on the cloud, with an example.

3. THEORY OF CONSTRAINTS

Theory of Constraints (TOC), a management philosophy developed by Goldratt (1990, 1997), is a theory that “suggests that all systems are similar to chains—or to networks of chains” (Dettmer 1998, p.11). The reason for this is that systems are composed of interrelated and interdependent links working together to achieve a predetermined goal. To be stable, systems must have very few leverage points (referred to as constraints)—otherwise chaos will occur (Goldratt 1997). Since the strength of a chain is determined by its weakest link, one of the basic assumptions of TOC is that for a system to improve, it must focus on its weakest link (any process within an organization which prevents the system from achieving its goals). The idea then is to construct a description of a system in such a way that it is possible to determine the key leverage points and what type of intervention is needed to obtain a desired result.

The heart of TOC, at this point in its development, is a set of logical tools known as the thinking processes (TP). These tools—used to analyze complex systems—

are based on strict logical procedure and have been used in myriad business applications. TP tools include the current reality tree, the cloud, the future reality tree, the negative branch, the prerequisite tree, and the transition tree. Specific details about the TP are available in the literature (Dettmer 1997; Scheinkopf 1999). All TP tools can be used to teach business courses, but given the special characteristics of MIS classes discussed in the introduction, the tools adapt well to teach information systems courses. The reason is that the TP tools, designed to attack systemic problems, try to capture the interaction among different elements, managerial and technical. Covering all the potential applications of the TP to teach MIS is beyond the scope of this paper; the purpose here is to show one of the tools (the Cloud) and its applications to writing MIS mini cases.

4. CONFLICT RESOLUTION AND THE CLOUD

One of the key elements for successful cooperative learning is developing the ability of students to use social and small group skills such as conflict management (Lancaster and Strand 2001). Tools that can help students deal systematically with conflicts can enhance group interaction in cooperative learning in several ways. First, conflicts can occur among team members for different reasons; some of these conflicts may come from some team members not putting their share into the effort. To resolve such conflicts, different techniques such as individual quizzes and tests have been suggested (Siciliano 2001). The other context, the focus of this paper, involves including conflicts within the mini-cases.

In TOC, the Cloud, a thinking tool used to represent conflicts (Scheinkopf 1999), has been used in many business situations as a problem-solving technique (Smith 2000). The Cloud of TOC can be used as a framework to include the essential elements in a conflict in a mini-case. This framework suggests that any conflict should include a minimum of five elements: a Common Objective (A), two Needs (B and C), and two Wants (D and D'). The structure of the Cloud can be seen in Figure 1. A Common Objective (A) represents

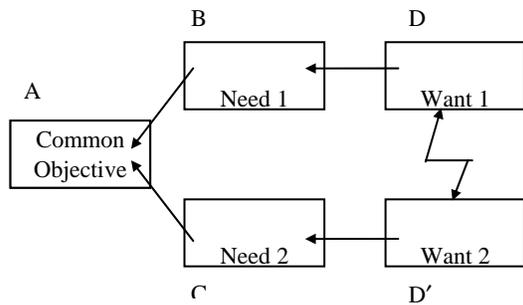


Figure 1. The Cloud

something that both parties in the conflict agree is important. At least two Needs (B and C) are required to achieve the Common Objective. Within a conflict, each side “owns” a Need. However, one of the sides in the conflict argues that in order to have Need 1 (B), he/she must have Want 1 (D). The other side insists that in order to have Need 2 (C), he/she must have Want 2 (D'). The conflict comes from the fact that D and D' cannot coexist

The Cloud offers the essential components students need to come up with creative solutions to solve a mini-case. First, it clearly shows the root of the conflict represented by the two Wants. Moreover, information is given as to the reason why each side insists on what they want. Also, it shows that there is a good reason why the conflict needs to be resolved. Thus students can focus on ways to find potential win-win solutions that satisfy the needs of both parties.

5. USING THE CLOUD TO WRITE A MINI-CASE

Most good managerial cases involve some type of conflict or dilemma (Stringer 1999) (e.g., buy vs. not buy, invest vs. not invest, centralized vs. decentralized).

In the case of MIS, technological solutions have been developed to solve or deal with some of these managerial conflicts. Mini-cases based on the Cloud can be used to encourage discussion, and help students develop the concepts themselves. Instructors expect students to recommend actions that will solve the problems presented in the cases, using knowledge either gained in class or acquired in their workplace.

To integrate the cloud into a class, the author uses the following steps:

1. Decide on the concept to be taught.
2. Find a relevant cloud and write a mini case.
3. Give the mini case to interdisciplinary groups for discussion.
4. Ask students to decide what the problem is and what “features” a technology should have to resolve the dilemma.

In step 1, the instructor decides what type of technology a student will most need to learn. Examples of technologies the author has taught using this procedure includes thin clients, data mining, and virtual private networks. Step 2 will be discussed with more details in the following section. One of the important issues regarding step 3 is ensuring that the teams are interdisciplinary, to obtain the full benefits of cooperative learning. At the end of the discussion, groups can present their wish list, step 4, as to what type

of technology they think will solve the conflict. The instructor finishes up the session by expanding upon the students' findings.

6. PROCEDURE TO WRITE A MINI-CASE

The Cloud of TOC provides a framework for how to write a mini-case. The mini-case should include the five elements from the Cloud. In addition, the mini-case should be written so that it can be solved using the technological solution we are trying to teach. By using a mini-case, students can "invent" a variety of academic content which should promote better understanding as well as more retention of the material. To illustrate the procedure, let's use as an example the MIS concept of expert systems.

6.1 Developing a Concept Cloud

First of all, we need to develop a Cloud to serve as the basis for writing the mini-case. Here are the steps:

1. Select a typical problem the technology is supposed to address. For example, if we were teaching the concept of expert systems, a typical problem could be this: a company is experiencing a long processing time.
2. Make the problem found in step 1 as specific as possible. Many times when step 1 is completed, the resulting problem is generic and may not be useful to write a realistic mini-case. Continuing the example about expert systems, we could come up with something more specific: customers of an insurance company have to wait too long for underwriters to make a decision on their insurance applications. Notice that the problem includes some type of business-related issue, namely customer satisfaction. Ideas for step 2 can come from reading professional literature on the topic or visiting organizations that experienced similar problems.
3. Write the opposite of the specific problem from step 2. (In the concept cloud, Need 1 (B) is the opposite of the specific problem found in step 2.) For our example about expert systems, B can be "customers know the result of their applications in a few days."
4. Find an action to achieve B that is not as good as the concept we are trying to teach. This action will be our Want 1 (D). This is an important step, because the solution we are teaching must be a potential improvement over the alternative selected in this step. How do we know our solution is better? The concept we are teaching should be more efficient in terms of time, cost or customer satisfac-

tion. For the expert system example, D could be "hire more underwriters," which could be more costly in the long run than having an expert system. Other alternative actions that could be used as Ds include "accept applications from fewer customers," or "send customers to a competitor."

5. Write Want 2 (D') as the opposite of D. As mentioned in step 4, D achieves our Need 1, but it is not as "efficient" as the solution we are trying to teach. As a result, the mini-case is presented as a dilemma between D and D'. In our example, D' is simply "do not hire more underwriters."
6. Write the reason why you may want D'. In our example, hiring more underwriters could result in a substantial increase in personnel cost, which is undesirable. As a result, the reason for wanting D' (do not hire more underwriters) is Need 2 (C), which can be written as "keep expenses under control."
7. Find the Common Objective (A). To find A, we ask: what is accomplished if needs B and C are met? In our example, we could say that A is "has a profitable business." Both needs—reducing lead-time and keeping expenses under control—are prerequisites to ensure that a business is successful. Other common objectives that can be used include "improving the atmosphere of the organization," or "having an effective department."
8. Refine the Cloud and use it as a basis to write the mini-case/story. Make sure the two Needs are clear prerequisites to attain the Common Objective. Also, check that there is a logical connection between the Needs and the Wants. Overall, make sure the conflict makes sense and is clear enough

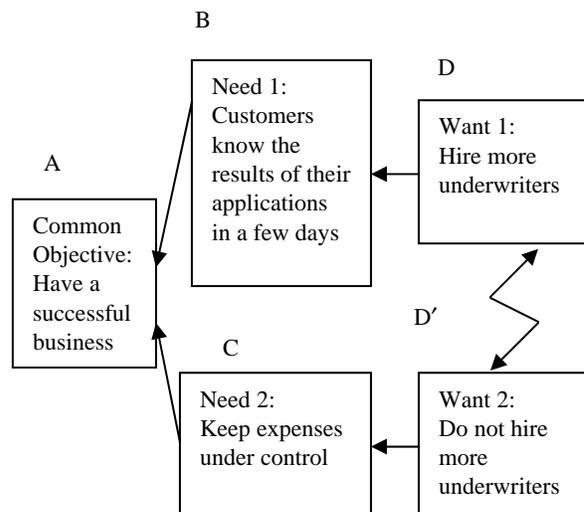


Figure 2. Expert System Cloud

so that students can use it to generate the solution(s) you are trying to teach. See Figure 2 for the expert systems' Cloud.

6.2 Writing the Mini-case

The previous steps provide enough data to write a very focused mini-case which students could debate in small groups during a regular class session. Writing a mini-case is a matter of style rather than a specific methodology, but the following are some guidelines that can be considered.

1. The introduction to the mini-case can be written by talking about a fictitious company to give some sense of reality to the case. Ideas to pick a type of company can come from the textbook, articles from professional journals, and visits to companies.
2. After a type of company is selected, the next part of writing the mini-case is to select what department or departments within the organization will be on what side of the conflict. This will depend on the Wants written in the cloud. In the example presented in this paper, one side of the conflict is the department that wants to hire more underwriters. Most likely this will be the department that is in charge of both analyzing the applications and making sure they are completed in a timely manner. Another potential ally of this side is customer service and the agents selling insurance. The other side of the conflict is the one that does not want to hire more underwriters. Most likely, this will be the finance department or the HR department.
3. Finally, and once the two sides have been selected, the mini-case needs to be written, making sure that all the elements of the cloud—Wants, Needs and Common Objective—are included.

Let's see a possible scenario:

John Rakow, head of the underwriter department at Frankem Insurance, has been swamped over the last years with complaints from customers. About 30% of the customers think that the time it takes to process an insurance application is too long. The situation is getting even more delicate as he has learned that some insurance agents do not want to recommend Frankem services (even though their prices are competitive) because their smaller competitors are able to respond faster. John is presenting his case to Alice Smith for a third time. Alice is in charge of making personnel decisions. John thinks that he needs to increase his department's personnel by at least 35% to cope with the current demand. Alice clearly understands John's

concerns but she has made some projections and according to her calculations, the expected benefits do not justify hiring more underwriters. She feels that the department will increase expenses without a good return on investment. Both want the business to be successful, but a creative solution is needed to ensure that objective. Can you help John and Alice?

7. CONCLUSION

The author has used this procedure to write mini-cases very successfully. Several objectives are accomplished when using mini-cases. First, students who are computer-oriented get a better idea about how technology fits within a business environment, including some of the "people issues" often ignored. Students' comments from the instructor's evaluation have confirmed this. One student wrote, "I liked ... how he attached business cases to IT [information technology] problems, and [then] presented [how] new technologies solve the problems." Also, students have a chance to "test" their technical knowledge in practice and share with their team members. Business-oriented students learn how to analyze situations that can be solved with technology, and in the process, they interact with other students who may have a different perspective and expertise. In many cases, students come up with different solutions to the problem posed on the case, making the environment fertile for discussion and debate. This cross-pollination can be very beneficial (Whipple 1987). A side benefit includes learning to work in teams, a very important skill in today's business environment.

8. FUTURE RESEARCH

This paper describes only one aspect of using the thinking processes of TOC to teach MIS classes—how to write a mini-case. However, further research is needed to determine how to fully integrate TOC with cooperative learning in the classroom. As suggested by Siciliano (2001), cooperative learning includes more than just putting students in groups. There are other elements, such as how to ensure individual accountability and improve team interaction (Siciliano 2001), which can be included as part of the research protocol.

Empirical research is also needed to test the benefits of using the Cloud process to write mini-cases and to assess the advantages and disadvantages of using TOC as a strategy to enhance cooperative learning. The empirical research can be extended to other subjects within the information systems area, such as electronic commerce, and system analysis and design. Since other business courses also use cases, a research project could include areas such as marketing, accounting, and operations management.

9. REFERENCES

- Dettmer, W. [1997], Goldratt's Theory of Constraints. ASQC Quality Press, Milwaukee.
- Dettmer, W. [1998], Breaking the Constraints to World-Class Performance. ASQC Quality Press, Milwaukee.
- Fellers, J.W. [1996], "Teaching teamwork: Exploring the Use of Cooperative Learning Teams in Information Systems Education." DATA BASE, Vol. 27, No. 2, Spring, pp. 44-60.
- Goldratt, E. [1990], What is this thing called the Theory of Constraints? Great North River Press. Barrington.
- Goldratt, E. [1997], Critical Chain. North River Press, Great Barrington.
- Granger, M.J. and Susan K. Lippert [1999], "Peer Learning across the Undergraduate Information Systems Curriculum." The Journal of Computers in Mathematics and Science Teaching, Vol. 18, No. 3, Summer, pp. 267-285.
- Jacobs, F.R. and D. Clay Whybark [2000], Why ERP? A Primer on SAP Implementation. McGraw Hill, Boston.
- Johnson, D., Roger T. Johnson and Karl A. Smith [1998], "Cooperative Learning Returns to College: What Evidence is there that it Works?" Change, Vol. 30, No. 4, July/August, pp. 26-35.
- Lancaster, A.K. and Carolyn A. Strand [2001], "Using the Team-Learning Model in a Managerial Accounting Class: an Experiment in Cooperative Learning." Issues in Accounting Education, Vol. 16, No. 4, November, pp. 449-567.
- Leidner, D.E. and Sirkka L. Jarvenpaa [1995], "The Use of Information Technology to Enhance Management School Education: A Theoretical View." MIS Quarterly, Vol. 19, No. 3, September, pp. 265-291.
- Mukherjee, A. and John Cox [2001], "Using Electronic Quizzes to Promote Self-Reliance in Minicase Analysis in a Decision Support Systems Course for MIS Majors." Journal of Education for Business, Vol. 76, No. 4, March/April, pp. 221-225.
- Newstrom, J.W. and Mark L. Lengenick-Hall [1991], "One Size Does Not Fit All." Training & Development, Vol. 45, No. 6, June, pp. 43-48.
- Scheinkopf, L.J. [1999], Thinking for a Change. St. Lucie Press, Boca Raton.
- Shlechter, T.M. [1990], "The Relative Instructional Efficiency of Small Group Computer-Based Training." Journal of Educational Computing Research, Vol. 6, No. 3, pp. 329-341.
- Siciliano, J.I., [2001], "How to Incorporate Cooperative Learning Principles in the Classroom: It's more than just putting students in teams." Journal of Management Education, Vol. 25, No. 1, February, pp. 8-20.
- Slavin, R.E. [1990], Cooperative Learning: Theory, Research and Practice. Prentice Hall, Englewood Cliffs.
- Smith, D. [2000], The Measurement Nightmare. St. Lucie Press, Boca Raton.
- Stringer, D. [1999], "Case Writing 101." Training & Development, Vol. 53, No. 9, September, pp. 52-57.
- Whipple, W.R. [1987], "Collaborative Learning: Recognizing it When We See it." Bulletin of the American Association for Higher Education, Vol. 40, No. 2, October, pp. 3-7.

AUTHOR BIOGRAPHY

Danilo Sirias is an assistant professor of Management at Saginaw Valley State University. His latest research relates to using the thinking processes of the Theory of Constraints as pedagogical tools. He has published in several journals including the *Journal for Quality and Participation*, the *International Journal of Applied Quality Management*, and the *International Journal of Production and Operation Management*. He is also a co-author of the book "Bridging the Boomer Xer Gap," published by Davies-Black.





STATEMENT OF PEER REVIEW INTEGRITY

All papers published in the Journal of Information Systems Education have undergone rigorous peer review. This includes an initial editor screening and double-blind refereeing by three or more expert referees.

Copyright ©2002 by the Information Systems & Computing Academic Professionals, Inc. (ISCAP). Permission to make digital or hard copies of all or part of this journal for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial use. All copies must bear this notice and full citation. Permission from the Editor is required to post to servers, redistribute to lists, or utilize in a for-profit or commercial use. Permission requests should be sent to the Editor-in-Chief, Journal of Information Systems Education, editor@jise.org.

ISSN 1055-3096