International Journal of Production Research, 2013
Vol. 51, No. 8, 2505–2517, http://dx.doi.org/10.1080/00207543.2012.743687

A managerial decision-making web app: Goldratt's evaporating cloud

Soeren Andersen[a]*, Mahesh Gupta[b] and Ankush Gupta[c]

[a]*Department of Entrepreneurship and Relationship Management, University of Southern Denmark, Kolding, Denmark;* [b]*Department of Management and Entrepreneurship, University of Louisville, Louisville, KY, USA;* [c]*Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA, USA*

(*Received 8 February 2012; final version received 11 October 2012*)

This paper presents a web-based interactive application designed to complement and expand upon a relatively new conflict elimination tool, the evaporating cloud, introduced by Goldratt in the business novel *It's Not Luck* (Goldratt, E., 1994. It s not luck. Croton-on-Hudson, NY: North River Press). The application allows users to enter information concerning a conflict in a step-by-step manner providing opportunity to revisit input entered in previous steps. It presents the conflict in a simple, logical diagram, and systematically guides the user through the process of surfacing underlying assumptions and developing potential solutions. Based on our experience in a senior-level operations management course, we find that most students consider the web app useful in helping them articulate and resolve conflicts and, thereby, make good decisions. Suggestions for using the application in an operations management class are included.

Keywords: decision making; conflicts; evaporating cloud; web app; active learning

1. Introduction

In today's constantly changing and turbulent global economic environment, organisations are increasingly being reorganised as supply chain network members and are adopting matrix structures where employees from different echelons are working in virtual teams. These changes are blurring the reporting lines and causing managers across functional areas to reengineer their organisations by ''identifying and abandoning the outdated rules and fundamental assumptions that underlie current business operations'' (Hammer and Champy 1993, p. 3). Increasingly, managers are finding themselves engaged in managing differences and addressing conflicts affecting their ability to make good decisions (Cloke and Goldsmith 2005, Gilboa 2011).

From an operations management (OM) perspective, most OM problems increasingly span organisational boundaries, and managers are frustrated in employing OM tools, techniques, and theories because certain factors that influence human behaviour such as decision biases, emotions, and culture are not considered explicitly (Bendoly *et al*. 2006). Recognising that the human element is a critical element in operations management, Loch and Yaozhong (2005, p. 15) issued a call for behavioural OM research which can be broadly viewed as an ''opportunity to truly reflect upon some of the long held assumptions on which much of operations research models have been founded.'' Recently, Gino and Pisano (2008) asserted that behavioural OM research involving people issues might lead to a better understanding of puzzling pathologies, for example, excess inventory and the efficiency myth. One of the key questions in behavioural OM is: how can managers systematically surface the assumptions underlying a conflicting situation and evaluate its validity?

Most academicians who have used the business novel *The Goal: A Process of Ongoing Improvement* (Goldratt and Cox 1984) in their operations management courses would agree that its underlying business plot epitomises the above discussion. It is a story of a manufacturing plant requiring heavy machinery and elaborate production processes as well as interactions among human resource, accounting, sales, and marketing personnel at corporate/regional headquarters. By using the concepts and tools of the theory of constraints (TOC), authors challenge long held assumptions underlying various concepts, such as the use of performance incentives, product cost, constraint optimisation, economic order/production quantity, local efficiency, and productivity measures. From this popular story, it becomes abundantly clear that organisations that fail to surface assumptions underlying workplace conflicts run the risk of making bad decisions and losing their competitive advantages.

*Corresponding author. Email: soa@sam.sdu.dk

In a relatively new senior-level undergraduate OM course, we experimented with a very general decision-making, or conflict elimination tool[1], the evaporating cloud, based on the theory of constraints as explained through *The Goal* (Goldratt and Cox 1984) and another of Goldratt's business novels, *It's Not Luck* (Goldratt 1994). In these novels, authors demonstrate that, in order for an organisation to improve its financial performance, it must focus its efforts by continuously asking three fundamental questions: (1) what to change (that is to say, what is the core problem?), (2) what to change to (that is, what are simple win–win solutions?), and (3) how to cause the change (namely, how to implement the proposed solutions?). To answer these questions, Goldratt (1994) proposes a set of techniques collectively known as thinking processes (TP) which can be used as a set of integrated tools or as stand-alone tools to address specific aspects of a problem. In our teaching experience, students see the benefit of mastering these tools but find them very challenging to learn. Many colleagues with whom we have spoken at conferences over the years share similar experiences. We believe that an interactive application that leads users through the process of developing the thinking process tools would help address the steep learning curve we have seen in practice.

In this paper, we present a web-based interactive application designed to complement and expand upon one specific TP tool: the evaporating cloud (EC). Goldratt (1990) introduced the EC as a means of developing simple win–win solutions to the long-standing conflicts in manufacturing management. Although there is a growing community of EC users, it is still a relatively new decision-making approach. This paper shows how this web app helps in making better decisions by systematically (1) framing each problem as a conflict, (2) surfacing underlying assumptions, and (3) developing solutions that invalidate one or more of the assumptions. Our classroom experience suggests that a web app that leads users through the development of the EC will make it easier for both managers and students to learn and apply in real-world situations.

The rest of the paper is organised as follows: first, we explain the structure of the evaporating cloud and briefly review the pertinent literature. Next, we describe a course challenge to explain how exactly the evaporating cloud is employed in the class. We also share an example decision situation from *The Goal* that is then used in the rest of the paper. In the section thereafter, we discuss the development of a web app following the guidelines suggested in the reviewed literature. We discuss its application in a class and feedback from the students. Finally, we conclude our paper with some possible future research directions.

2. The structure of evaporating clouds

The evaporating cloud is composed of five entities structured as shown in Figure 1. As mentioned above, Goldratt believes that most problems faced by managers can be viewed as conflicts, either between two parties (people or departments) or, frequently, internal conflicts experienced by an individual. What one party to the conflict wants is

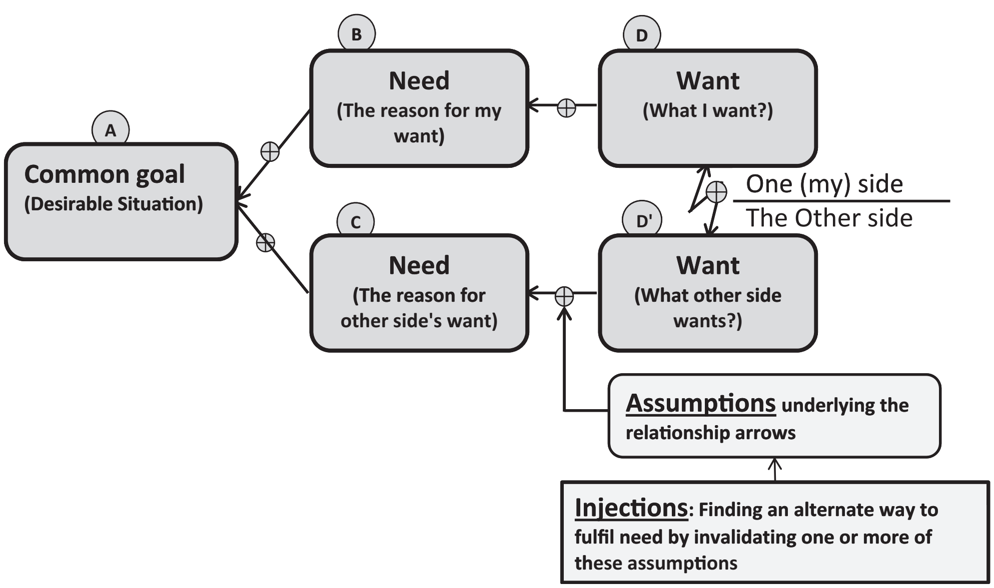


Figure 1. Structure of an evaporating cloud.

entity D and what the other party wants is entity D′. These two entities must be in conflict either because they are mutually exclusive or due to resource contention, that is to say, the organisation cannot afford to do both.

The structure of a cloud shows that each party's want is necessary in order to satisfy a specific need denoted by entities B and C. In addition, both needs must be met in order to achieve the parties' common goal, denoted by entity A. In other words, these two needs are necessary conditions for accomplishing the common objective.

Underlying each of the arrows in the EC is one or more assumptions explaining the conditions under which the relationship between two entities in the cloud is valid. Assumptions underlying arrow C–D′ in Figure 1 explain why D′ is a necessary condition in order for the need C to be met. In the event that a necessary assumption under arrow C–D′ can be rendered invalid, D′ will no longer be a necessary condition for achieving need C. By removing D′ as a necessary condition for C, the conflict between D and D′ is eliminated. In the EC, assumptions are statements about reality that are accepted as true even if the statement is untested. One way to invalidate an assumption is thus to provide evidence that the assumption is not valid, that is, that the entity at the base of the arrow is not actually necessary in order to have the entity at the head of the arrow. When the assumption is valid, another approach is to come up with an action or change in conditions (referred to as an injection) that will make the assumption invalid.

Assumptions under the double headed arrow between D and D′ explain why the two wants are mutually exclusive. Invalidating assumptions under this arrow makes both D and D′ possible simultaneously. Surfacing assumptions and coming up with injections is not equally difficult for all arrows in a cloud. The arrows between the objective (A) and the needs (B and C) are typically fundamental (that is to say, hard to prove invalid) assumptions. When the relation between A–B or A–C is broken, D or D′ is no longer a reasonable action (along with all other actions the company takes to fulfil the now invalid need). The relations between B and D and C and D′ are usually the easiest place to surface assumptions and develop injections. However, the most powerful solution is found between D and D′, but the assumptions here are probably also the most challenging to invalidate. The end result of this process of analysing the cloud should be at least one feasible injection that invalidates an assumption and breaks an arrow between any two entities in the cloud (Goldratt 1990).

In the first academic paper concerning the EC (Jackson *et al.* 1994), the authors end by warning the reader about the difficulties of learning the technique. Around the same time, Goldratt (1994) published *It's Not Luck*, a sequel to *The Goal*, which illustrated the usefulness of evaporating clouds in resolving a variety of inter-organisational, intra-organisational, inter-personal, and intra-personal conflicts. Since then, several authors have provided guidance on the process of constructing clouds in the form of books (for example, Scheinkopf 1999, Dettmer 2007, Cohen 2010), and journal articles (Dettmer 1999). Gupta *et al.* (2011) demonstrated that the evaporating cloud incorporates well-accepted principles of achieving win–win solutions, such as separating the people from the problem, focusing on needs and not on positions, and helping identify the assumptions blocking win–win solutions (Fisher and Ury 1982).

3. Course challenge

In the context of a required course project (worth 20% of the student's final grade) for a senior-level course, we ask students to demonstrate TOC applications in a real-world company. Throughout the course, while reading *The Goal,* students are instructed to pay close attention to the conventional OM concepts (for example, efficiency measurement, economic production/order quantity, resource optimisation, and so forth) and the related conflicts faced by the fictional plant manager, Alex Rogo. Although students are intrigued by the Socratic approach used in *The Goal* to question the validity of several assumptions made by Alex and his associates, the students find seemingly obvious TOC-based solution approaches difficult to implement at their respective workplaces. They note that initially even Alex faced resistance to change in the novel.

In this paper, we suggest that if the students are equipped with a structured approach such as the evaporating cloud to eliminate conflicts, they will be able to identify conflicts underlying their decisions and, importantly, be able to develop and implement win–win solutions to their unique problem situations. In our work with students, we have found that they have no problem understanding the basic idea behind the cloud as shown in Figure 1, but they struggle to develop clouds of their own. More importantly, they find it very difficult to surface assumptions and to find injections to invalidate them. To assist students and to make the EC accessible to the OM community, we developed a web-based interactive application. To illustrate our web app, we assume that Alex is confronted with the following situation from *The Goal*:[2]

In Chapter 11, on page 84 of *The Goal*, Alex meets with Jonah and boasts of the efficiencies of his plant, specifically one particular expensive piece of equipment. Efficiency is one of the primary measure executives at

UniCo use to measure plant performance. In order for his plant to be successful, Alex believes that workers must not be idle as idle workers are not productive and are therefore a bad thing.

Jonah, on the other hand, argues that, ''A plant in which everyone is working all the time is very inefficient'' (Goldratt, p. 84). According to TOC, if every business process produces as efficiently as possible, inventory and operating expenses will climb while throughput declines. Alex has to make a decision: idle the workers and face decreased efficiencies or keep them working and damage throughput.

In the next section, we use this situation to show how Alex might employ the web app to make the decision, and to illustrate our development process.

4. The process of evaporating clouds: Web app development process

Even though a cloud consists of only five entities, the difficulty of the process of verbalising a conflict as a cloud should not be underestimated. In our experience, surfacing assumptions is much harder when the entities of the cloud (D, D′, B, C, A) fail to capture the essence of the conflict. Although not important from the web app user's perspective, we now describe the process employed in the development by grounding it in the relevant literature and, thereby, we highlight its theoretical perspective.

Different authors recommend different approaches as to how one should construct an evaporating cloud[3]. Scheinkopf (1999) explains a process with three major steps which seeks to first articulate the conflict and diagram the cloud, uncover assumptions, and in the end develop injections to break the conflict. Dettmer (2007) suggest a process of nine steps to accomplish the same three stages of constructing the cloud, surfacing assumptions, and developing injections. Cohen (2010) suggests seven steps to cover the process. To synthesise these three different approaches, it seems appropriate to maintain three general stages: (1) describing, articulating, and validating the conflict, (2) surfacing underlying assumptions, and (3) developing injections to resolve the conflict.

Stage 1 begins with a description of the conflict, where the context of the problem is given. The next part of stage 1 is to articulate the problem as a cloud and validate it. Stage 2 then is about surfacing assumptions underlying the arrows between the entities of the cloud. In the final stage the assumptions are checked to see whether any of them are non-existing or can be made invalid.

Although the literature offers various guidelines to systematically construct a cloud and resolve it, we found that students continued to struggle to remember these guidelines and thus spent too much time on the process of getting the conflict verbalised. Without a clear verbalisation of the conflict, it becomes hard for the student to surface assumptions and in turn, harder for the course instructor to grasp the conflict. To remedy these issues, we developed a web app to aid students in framing their problems as clouds and then to guide them in eliminating the underlying conflicts. This purpose is achieved by taking the student through the three-stage process of articulating the conflict, surfacing assumptions, and developing injections, the starting point for a potential solution. Each stage of the process prompts specific questions after integrating users' previously provided input in a manner suggested in the literature.

4.1 *Stage 1: Describing the conflict*

Before initiating the three-stage process of constructing a cloud, the application asks users to briefly describe the conflict in everyday language. The description of the conflict establishes the context for the cloud and provides background information needed to understand the entities of the cloud by anyone who might be a party to the conflict or is trying to help in eliminating the conflict.

Both Scheinkopf and Cohen suggest that, before any attempt is made to diagram the cloud, the first step is to write a description of the conflict viewed with some distance. For Cohen, part of this exercise is to distinguish between different types of clouds, while Scheinkopf argues for this step both to elicit commitment towards a solution and to provide a wider perspective on the conflict. Dettmer's suggested process presumes that all relevant background information is either known or available and thus he is not preoccupied with this preliminary step.

The web app gives a simple example situation, the HR example, and refers users to a case study ''We Googled You'' which is available online for more details if needed (see Figure 2). It also provides tips, such as ''not to overanalyse the problem at this point'' and ''simply describe the problem from your perspective.'' (Coutu 2007).

For our purpose, we are assuming that Alex Rogo, the plan manager in *The Goal*, is using the web app to create his cloud. On the screen illustrated in Figure 2, Alex would name his decision situation, for example, an efficient

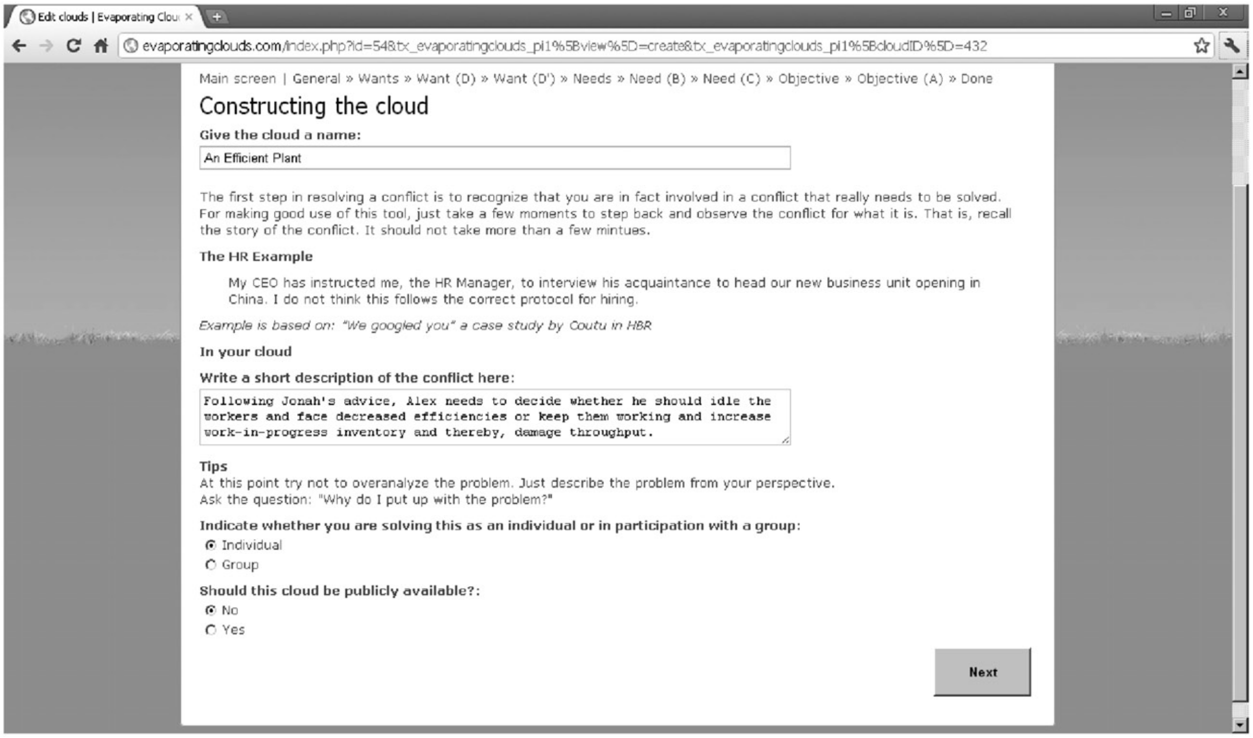


Figure 2. Web app screen shot to describe the situation.

plant, and briefly describe the decision he is faced with. In this case, his description might be *whether to idle the workers and face decreased efficiencies or keep them working and increase work-in-progress inventory and thereby, damage throughput*. The application provides an option of making this decision individually or as a group and saving the cloud in a private or public domain. The idea of having a library of clouds in a public domain is to allow potential users to see if there is an existing conflict situation that might resemble their own or to add one of their own clouds to the library. In addition, the application allows users to rate the likability of a cloud in the public library to indicate its potential usefulness.

4.2 *Stage 2: Articulating and validating the conflict*

At this stage, the application helps the user to think critically and articulate the conflict by identifying all its elements. The scholars cited above have suggested specific recommendations for coming up with the five entities of the cloud. While Scheinkopf does not give any preference for a starting point in the diagram, both Dettmer and Cohen suggest starting with the wants (the conflict, D or D′), followed by the needs (the necessary conditions of B and C) and finally the common goal (the objective A). Scheinkopf suggests the wants should be worded in such a way that the following statement would make sense, when D and D′ are substituted for real wants: *in order for* [D] *to exist,* [D′] *must not*. Dettmer suggests a similar statement: *on one hand we must* [D] *and on the other hand we must* [D′]. A similar method of replacing blanks is offered for the rest of the relationships in the cloud.

The web app provides an introduction to the wants (D and D′) as a quick refresher to outline what information is expected from the user. The first question the user is asked: "What action are you feeling pressure to take to deal with the problem?" The user answers the question in the following format: "I feel pressure to [D]", where [D] is simply an input-box in which the user inputs a phrase that makes sense in the context of "I feel pressure to". On the screen the user is provided with an example, the HR example, to show how the question may be answered in the context of the example conflict. In the situation we are describing from *The Goal*, Alex might answer the question as: "I feel pressure to: *keep workers busy at their workstations all the time*."

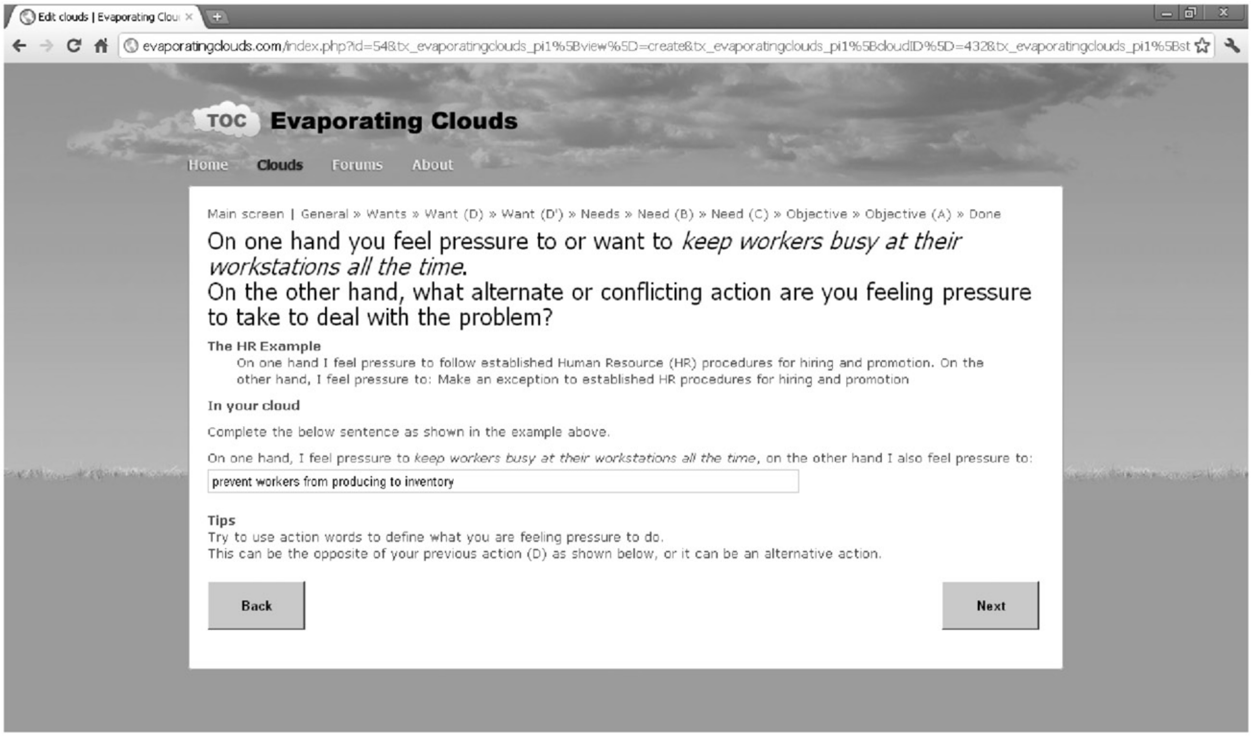


Figure 3. Web app screen shot of eliciting conflicting want D′.

Eliciting D′ is then done by integrating the answer for D in the question: "On one hand you feel pressure to (or want to) keep workers busy at their workstations all the time. On the other hand, what alternate or conflicting action are you feeling pressure to take to deal with the problem?" For our example situation, Alex might answer this question as: "on the other hand, I also feel pressure to: *prevent workers from producing for inventory*," (Figure 3).

The next step of this stage is to check the verbalisation. This is done by asking the user to read the following sentence out loud and judge its readability: "On one hand I feel pressure to *keep workers busy at their workstations all the time*; on the other hand, I feel pressure to *prevent workers from producing to inventory*." It is possible to go back and revise the wording or to continue with the needs (B and C). A short paragraph tells the user that the next couple of questions will concern the needs D and D′ are intended to meet.

The need B is then presented to the user (Figure 4). At this screen, the application again provides tips in the bottom of the screen to help the user choose a proper response. One of the tips is, "Alternatively, think of what need will you fulfil by deciding to [D]," and the answer should support [B]. In the example situation, "If I decide to *keep workers busy at their workstations all the time*, I will fulfil my need to [B]". This tip is the conventional way to elicit the needs B and C. However, since D potentially could satisfy different needs, the question posed to the user is aimed at getting exactly the need that is jeopardised by D′. The question follows the form: "What need will you endanger if you decide to [D′]?" The application asks the user to complete the sentence: "If I decide to [D′] I will endanger my need to [B]." For our example situation from *The Goal*, Alex might complete the sentence as: "If I decide to *prevent workers from producing to inventory* I will endanger my need to *have high efficiencies*." By doing it this way, a crosscheck, which is usually done after construction of the cloud (Cohen 2010), is integrated directly during the construction.

The process for the other need C is similar to the process for B. For the example situation, Alex might answer this question as: "If I decide to *keep workers busy at their workstations all the time*, I will endanger my need to *reduce work-in-progress inventory*".

At this point the user is again asked to validate the logical connections between the entities. This time the needs are shown in connection with the wants. In our example situation, Alex will validate the relationship between B and D by reading the following sentence out loud: "In order to *have high efficiencies*, I must *keep workers busy at their workstations all the time*." And the relationship between C and D′ is validated by reading this sentence out loud: "In

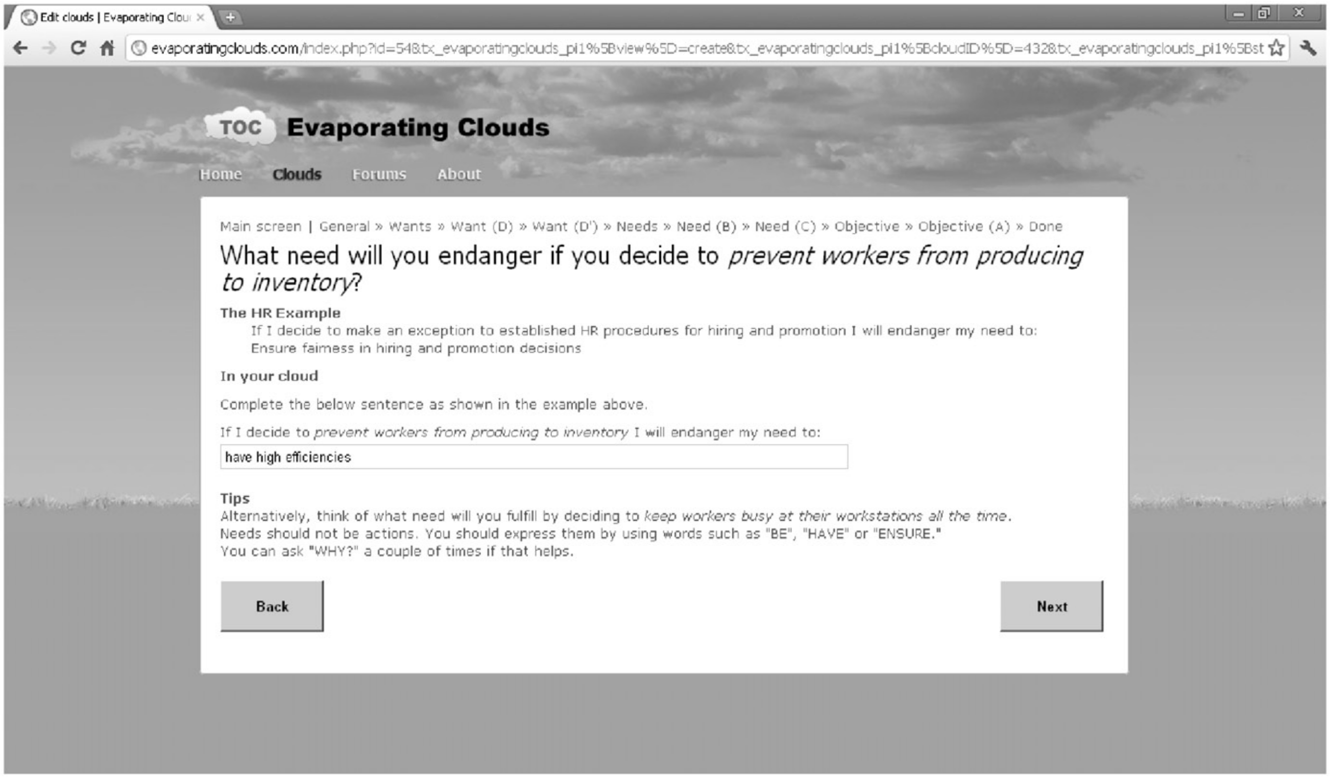


Figure 4. Web app screen shot of eliciting need B.

order to *reduce work-in-progress inventory*, I must *prevent workers from producing to inventory*.'' Again the user is encouraged to go back and revise if unsatisfied with these statements.

The last input collected from the user at the articulation stage is the common objective. Here the question integrates the two needs by asking: ''What objective will be fulfilled if both needs ([B] and [C]) are met?'' and the question is answered with: ''My objective is to [A].'' The question would look like this in the example: ''What objective will be fulfilled if both needs (*have high efficiencies* and *reduce work-in-progress inventory*) are met?'' The user is prompted to complete the sentence: ''My objective is to ...'' Alex, in our example situation, might answer: ''My objective is to *see that the Bearington plant makes a valuable contribution to UniCo*.''

The final step in the articulation stage is to validate that the cloud is in fact an accurate depiction of the conflict. In order to validate the cloud, sentences of the structure are read from left to right using the format: ''In order to [...], I must [...].'' For our example situation, the web app will instruct Alex to read the cloud: ''In order to *see that the Bearington plant makes a valuable contribution to UniCo*, I must *have high efficiencies*.'' ''In order to *have high efficiencies*, I must *keep workers busy at their workstations all the time*.'' ''In order to *ensure that the Bearington plant makes a valuable contribution to UniCo*, I must *reduce work-in-progress inventory*.'' ''In order to *reduce work-in-progress inventory*, I must *prevent workers from producing to inventory*.'' ''On one hand, I must *keep workers busy at their workstations all the time,* and on the other hand, I must *prevent workers from producing to inventory,* but I cannot have both simultaneously (or must pick one over the other).''

4.3 *Stage 3: Surfacing underlying assumptions*

Once all the elements of the cloud are identified and the cloud captures the essence of the conflict (Figure 5), the web app helps the user to surface and evaluate the assumptions underlying each arrow. To surface assumptions, Goldratt (1994), Scheinkopf (1999), and Dettmer (2007) suggest a sentence completion technique that involves adding the word ''because'' after reading each arrow. For example, for the B–D arrow we would read ''In order to [B], we must [D], because (for D and D′ the wording becomes: on the one hand I must [D], but on the other hand, I must [D′].

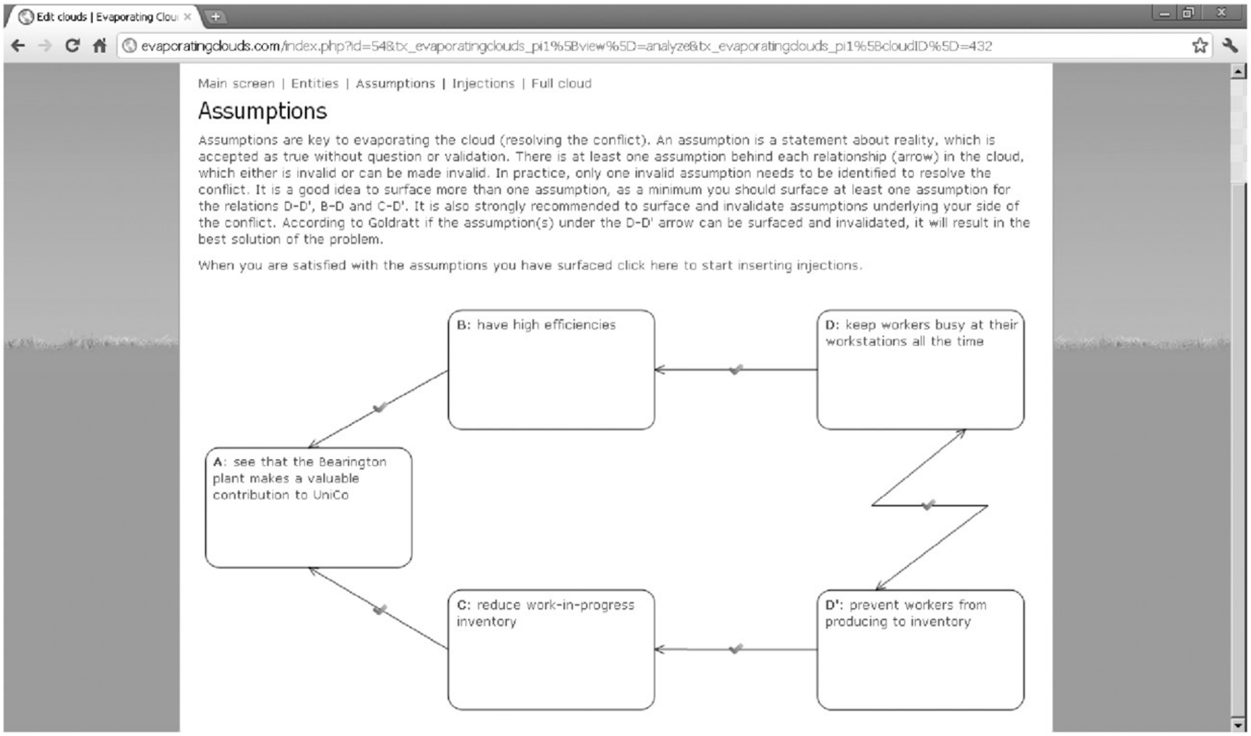


Figure 5. Web app screen shot of the complete cloud.

I cannot have both because). Dettmer further suggests that the use of extreme words (such as impossible, always, never) can provoke the mind to invalidate some assumptions and/or develop solutions. Scheinkopf and Dettmer also propose ways to check the quality of the assumptions. For example, Scheinkopf suggests that we should check whether the assumption is within the system under study and whether the assumption is valid. Dettmer, on the other hand, recommends that invalid assumptions should be marked with a symbol so that we can investigate them further in our search for a possible injection.

The web app makes use of these recommendations and helps the user surface the assumptions under each arrow. As shown in Figure 5, the application enables the user to click icons on any of the five arrows (A–B, A–C, B–D, C–D′, D–D′) between the entities. A click on any arrow brings up a screen as shown in Figure 6 where the user is asked a question structured in this way: "In order to [. . .] I must [. . .], because:" The application allows flexibility to surface assumptions under any arrows and as many assumptions as needed or desired. In general, we encourage students to surface at least one assumption under each arrow although generally several assumptions can be identified to come up with alternative solutions.

In the example situation from *The Goal*, Alex might challenge his side of the cloud first and surface at least two assumptions (Figure 6) as follows, "In order to *have high efficiencies*, I must *keep workers busy at their workstations all the time* because (1) *a worker who is not producing at his/her workstation lowers his efficiency*, and (2) *there is nothing a worker can do other than produce at his/her workstation to improve his efficiency*."

Once an assumption is surfaced, the application converts crosses (the Xs) on each arrow into checkmarks (the Vs), suggesting that at least one assumption under that arrow has been identified; thus, it can be used to further develop an injection in the next step.

4.4 *Stage 4: Developing injection to solve the conflict*

At this stage, the cloud should be fully developed with readable entities glued together by assumptions. The aim is now to find assumptions that can be invalidated. Both Scheinkopf (1999) and Cohen (2010) stress brainstorming and thinking out of the box at this point, meaning that ideas should be generated without initial judgment

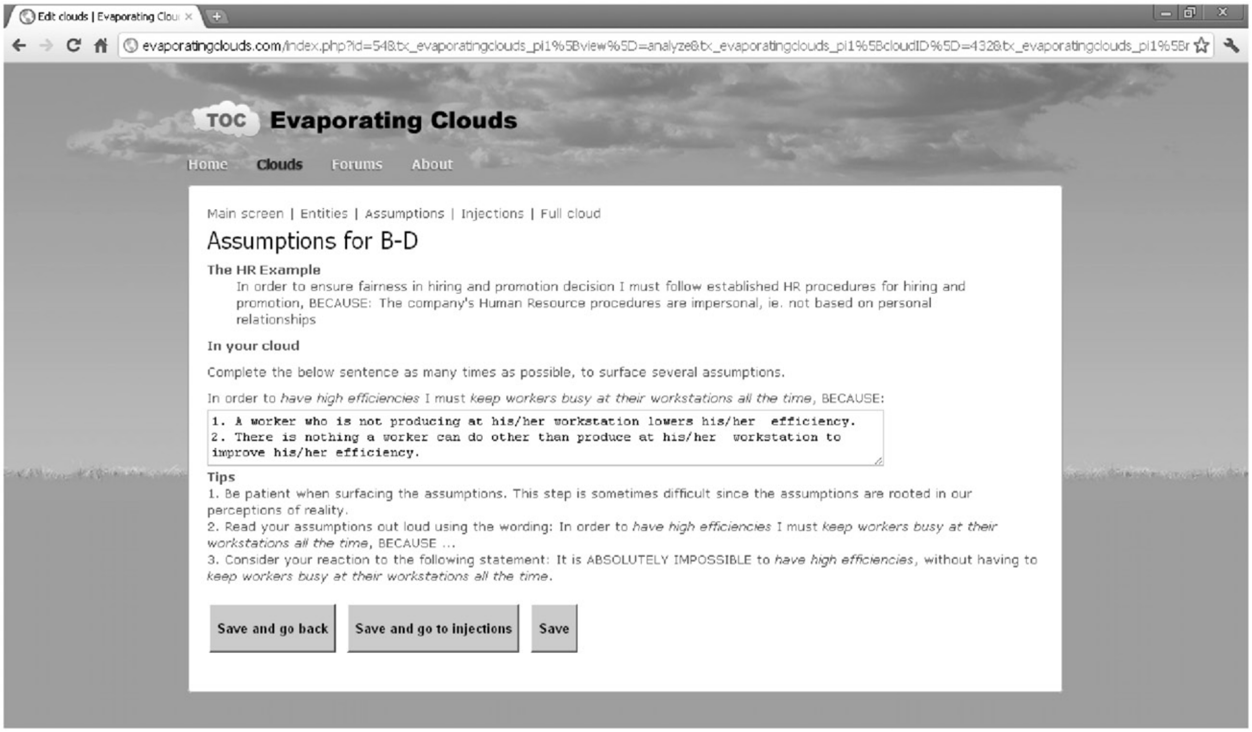


Figure 6. Web app screen shot surfacing the assumptions.

about feasibility. With regard to which of the five arrows (A–B, A–C, B–D, C–D′, D–D′) to attack in the diagram, the authors provide somewhat different suggestions. Scheinkopf advocates no arrow in favour of the other and thus implies that the solution could be found by attacking any arrow. Dettmer (2007), arguing that the needs may have been established under conditions that are no longer valid, advises the user to scrutinise the validity of the two needs B and C. Thus, Dettmer seems to suggest attacking the arrows A–B and A–C, before moving on to the other arrows. Cohen, although emphasising that assumptions under all arrows can be challenged, finds it most practical to focus on arrows between B–D, C–D′, and D–D′.

The injection screen of the web app is very similar to the assumptions screen (Figure 7). The application is very flexible as any assumption under any arrow can be chosen by the user to develop an injection. Although, theoretically speaking, for a cloud to be evaporated and conflict to be eliminated, only one assumption under one arrow needs to be invalidated, we encourage our students to find an injection for each arrow. The application provides tips to help users invalidate assumptions. For example, it suggests that the user asks himself/herself whether the assumption is always true? What example would demonstrate that the assumption is false?

Figure 7 shows a typical screen for coming up with injections. For example, in the situation from *The Goal,* Alex might challenge assumption 1 under arrow B–D (assumption 1: a worker who is not producing lowers his efficiency) and come up with an injection or an idea such as*: The efficiency measure could be changed so that the hours of an employee who does not have work to do are excluded from the efficiency calculations.*

Similarly, the application allows the user to let his/her creative juices flow and generate as many injections (possible solutions) as possible (Figure 8). Once the desired number of assumptions and injections is reached, the web app allows the user to save the cloud in PDF format for ease of sharing. The PDF version is complete with the description of the conflict situation, the cloud itself, the verbalisation of the cloud, and the assumptions and injections table (Figure 8).

5. Application in class

Evidence suggests that many students – especially non-OM majors – have difficulty maintaining interest in required OM core courses. Many OM scholars have argued in favour of active learning where students participate in an

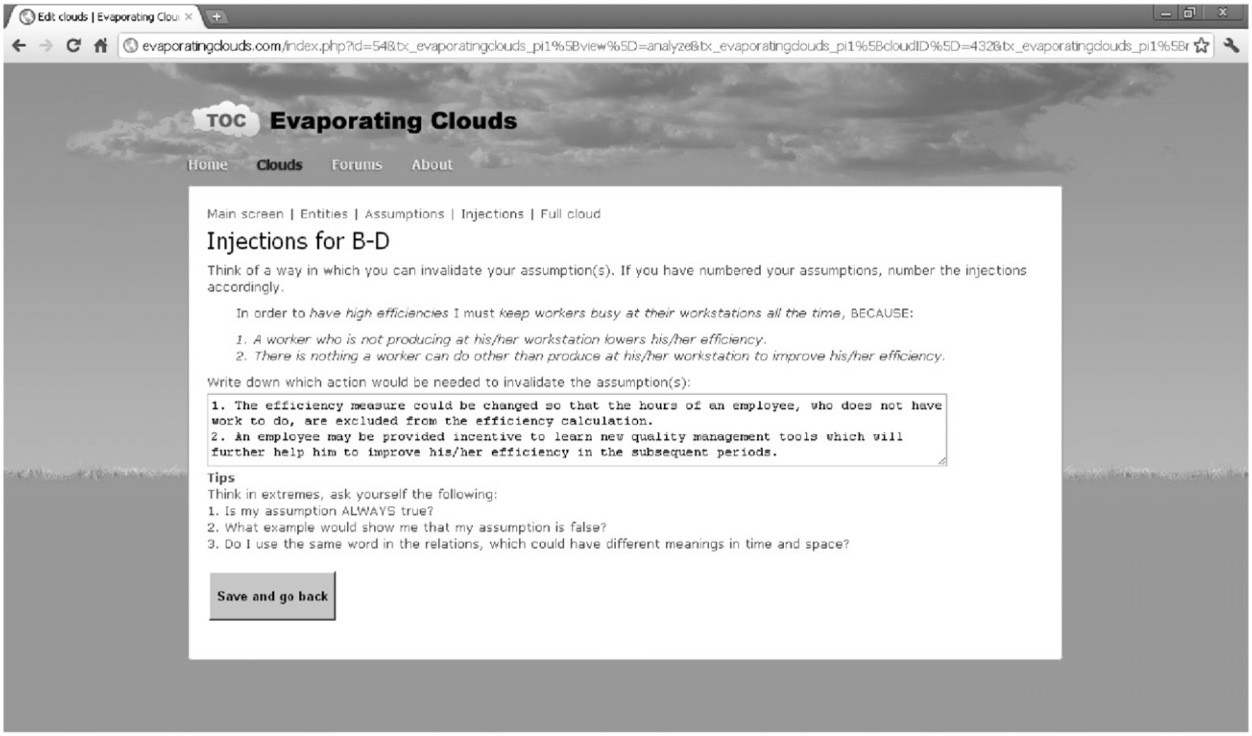


Figure 7. Web app screen shot developing an injection (solution).

experiential exercise, game, or hands-on simulation to master the key OM concepts (Heineke and Meile 1995, Umble and Umble 2005). Arguably, *The Goal* is the most interesting OM textbook because it provides novice students an easy-to-read, fictional case study showing how the realities of a complex manufacturing environment impact a manager's ability to make good decisions (Dani 2006). It also provides a real-world context to discuss principles of TOC as well as conventional OM concepts. Recently, Gupta and Boyd (2011) showed how to integrate *The Goal* together with a popular experiential exercise, the dice game, across OM topical coverage in a typical 15 week course.

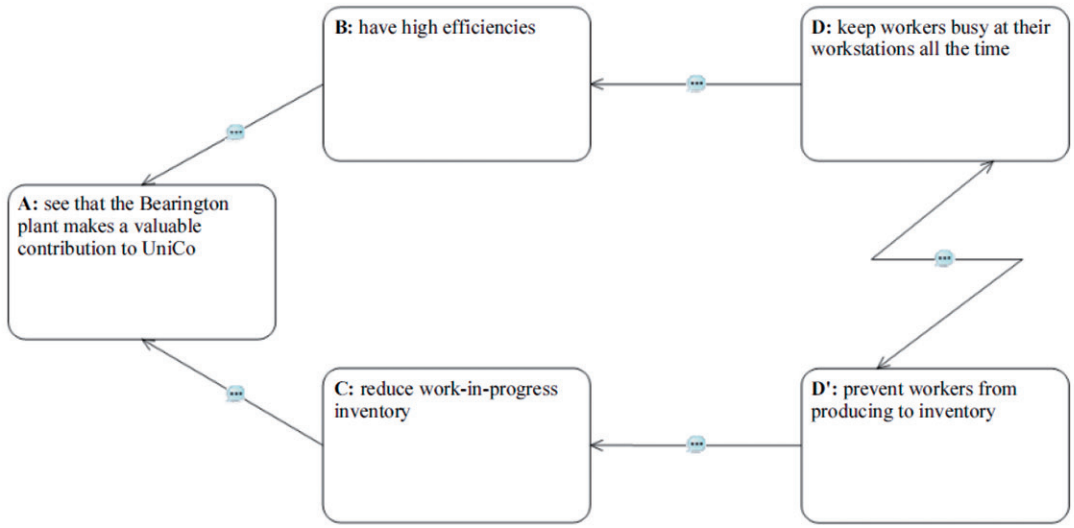
When working on their required course project, it becomes abundantly clear to our students that the managers are wasting their energy on the wrong policies, resources, and symptoms very much like the company discussed in the novel. In our experience working with students on their projects, as much as students feel pretty confident in their abilities to identify constraints and to come up with workable simple solutions, they find themselves up against the wall challenging their clients' usual ways of doing business, which are deeply rooted in their long-held beliefs and assumptions.

In this context, students are introduced to this web app. They are asked to identify a decision situation from *The Goal* and demonstrate how Alex could have used the web app to frame the decision as a cloud and arrive at the solution discussed in *The Goal*. We use the decision situation discussed in this paper as an example in the class and share the PDF copy of the cloud with students.

We have now experimented with this tool in our class in two ways. First, we ask students to identify two decision situations of their choice, preferably one from their work place and the other one from their personal or social life, if possible. Thus, in total we ask them to create three clouds starting with the one from *The Goal* and later using their personal and/or workplace-related decision situations. Second, we ask students to do a major course project demonstrating the application of TOC measures and five focusing steps for continuous improvement. We encourage students to recognise their want D, that is to say, what they feel pressure to do to solve the problem. Then D′, which represents current practice at their workplace, along with the respective needs (B and C), and a common objective (A), should be identified. After analysing the assumptions underlying the arrows, the students are expected to present an injection, which could serve as a starting point to eliminate the conflict. This approach clearly demonstrates their understanding of the existing situation and, importantly, the underlying assumptions.

An Efficient Plant

Following Jonah´s advice, Alex needs to decide whether he should idle the workers and face decreased efficiencies or keep them working and increase work-in-progress inventory and thereby, damage throughput.



In order to *see that the Bearington plant makes a valuable contribution to UniCo* I must *have high efficiencies* and in order to *have high efficiencies* I must *keep workers busy at their workstations all the time*. But, in order to *see that the Bearington plant makes a valuable contribution to UniCo* I must also *reduce work-in-progress inventory* and in order to *reduce work-in-progress inventory* I must *prevent workers from producing to inventory*. I can't both *keep workers busy at their workstations all the time* and *prevent workers from producing to inventory*.

Relation	Assumption(s)	Injection(s)
D-D'	1. The workers are kept busy producing even when there are no customer orders. Therefore preventing workers from producing to inventory and keeping workers busy are mutually exclusive.	1. Provide opportunities to get cross-trained to do other important jobs and thereby become valued employee. 2. Control the release of material as per the firm customer orders.
B-D	1. A worker who is not producing at his/her workstation lowers his/her efficiency. 2. There is nothing a worker can do other than produce at his/her workstation to improve his/her efficiency.	1. The efficiency measure could be changed so that the hours of an employee, who does not have work to do, are excluded from the efficiency calculation. 2. An employee may be provided an incentive to learn new quality management tools which will further help him/her to improve his/her efficiency in the subsequent periods.
C-D'	1. There is excess capacity on the workstations.	1. Remove excess capacity from the workstations without any negative consequences. 2. Work with market department to sell the excess capacity by introducing new features (at a price) or just exceed customer satisfaction to build a strong customer base.
A-B	1. Highly efficient workstations are considered a key performance measure by the top management. 2. Highly efficient workstations result in high return-on-assets.	1. Make top management aware of the dangers of using efficiency as a key performance measure. 2. Make top management aware of the long-term dangers of measuring local efficiencies of the workstations.
A-C	1. Work-in-progress inventory ties up cash. 2. Work-in-progress inventory lengthens lead time. 3. Work-in-progress inventory hides quality problems.	

Figure 8. Web app screen shot depicting the PDF copy of the complete cloud.

More importantly, it provides them an opportunity to lay out the proposed solution along with their assumptions. We ask them to include at least three of these clouds in their major project.

6. Feedback

Currently, this web app is being hosted at www.evaporatingclouds.com to address workplace-related situations. A comprehensive library of clouds (developed by students and voluntarily kept in the public domain) is in progress at www.evaporatingclouds.com. Further, the application has been used in a project focusing on the resolution of conflicts targeted at teenagers (Gupta 2010), and it has been deployed to a company intranet for use among managers in a Danish manufacturing company. A workshop focusing on the use of the web app was conducted at the 2010 annual meeting of the Decision Science Institute.

A Likert-style survey of 75 students was done to determine (1) the ease with which they were able to use the web app and learn how to operate the program, (2) the extent to which the steps of the program were clear and the program helped them to understand and resolve the conflict, and (3) the overall usefulness of the program itself. On a scale of 1–5, the students responded positively to all three questions (scores ranged between 4.16 and 4.833). In an open-ended question, an overwhelming number of students indicated that the web app supplements their intuitive approach with a deliberative approach to problem solving and decision making. Overall, our classroom experience shows that this application is viewed by students as valuable to their ability to resolve conflicts.

In addition, students easily relate to situations discussed in the public domain and point out similar conflicts they face at their workplace. We challenge our students to consider developing clouds (or even modifying existing clouds in the online library) to reflect conflicts and dissent among peers as well as superiors and find customised solutions to their unique problems.

7. Conclusion and future research direction

In this paper, we discussed a web-based application of the evaporating cloud. Students can employ it to resolve workplace conflicts, undertake organisational development interventions, and, importantly, initiate a change management process. We encourage students to revisit various decisions presented in *The Goal* to build their confidence and then to use the application to address problems encountered at their work places. From an academic perspective, the web-based application described in this paper addresses a shortcoming of the TOC thinking process tools, namely, that the thinking process tools are difficult to master and there is a need to make them more user-friendly, as pointed out by Watson *et al*. (2007).

A potential future research direction would involve checking the feasibility of the suggested solution by using the other tools of the TOC thinking processes such as future reality tree and negative branch reservation (Scheinkopf 1999, Dettmer 2007). Such an extension should be explored either as a stand-alone web app or as a part of this particular web app. Once the database of website users is established, a questionnaire can be administered to empirically compare conventional and TOC-based approaches to problem solving. The idea of integrating clouds within the framework of the TOC-based continuous improvement process is also worth exploring as a possible future project. Last but not least, efforts should be made to extend the application to a relatively lesser-known TOC tool known as the three-cloud approach.

Acknowledgements

The authors wish to express their gratitude to two anonymous reviewers and Dr Lynn Boyd, professor of operations management, for their insightful comments and many significant improvements to the text.

Notes

1. In the paper, "conflict elimination" is used purposefully instead of conflict resolution, because the aim of the evaporating cloud is different from most conflict resolution tools in that it always seeks a win–win solution instead of a compromise.
2. Since the evaporating cloud is based on necessity logic, it is possible to come up with different clouds for the same situation, depending on which needs are in focus. The example from *The Goal* is a subset of a larger discussion concerning measures, and attending to the overall discussion would result in a different cloud. In addition, while the example is based on a

situation from *The Goal*, the assumptions and injections are largely the author's ideas about how Alex might use the web app.

3. Twenty years have passed since Goldratt (1990) presented a cloud, the structure of the evaporating cloud is basically the same, but different advice on how to ease the process has emerged over time. The reader is referred to Cohen (2010), an article in the *Theory of Constraints Handbook*, for further information.

References

Bendoly, E., Donohue, K., and Schultz, K.L., 2006. Behavior in operations management: assessing recent findings and revisiting old assumptions. *Journal of Operations Management*, 24 (6), 737–752.

Cloke, K. and Goldsmith, J., 2005. *Resolving conflicts at work: eight strategies for everyone on the job*. San Francisco, CA: Jossey-Bass.

Cohen, O., 2010. Daily management with toc. *In*: J.F. Cox and J.G. Schleier, eds. *Theory of constraints handbook*. New York: McGraw-Hill, 671–727.

Coutu, D., 2007. We googled you. *Harvard Business Review*, 85 (6), 37–38.

Dani, S., 2006. Book Review: The goal – A process of ongoing improvement. *International Journal of Production Research*, 44 (2), 419–420.

Dettmer, H.W., 1999. The conflict resolution diagram: creating win–win solutions. *Quality Progress*, 32 (3), 41–47.

Dettmer, H.W., 2007. *The logical thinking process: a systems approach to complex problem solving*. Milwaukee, WI: ASQ Quality Press.

Fisher, R. and Ury, W., 1982. *Getting to yes*. New York: Penguin Books..

Gilboa, I., 2011. *Making better decisions: decision theory in practice*. Malden, MA: Wiley-Blackwell.

Gino, F. and Pisano, G., 2008. Toward a theory of behavioral operations. *Manufacturing & Service Operations Management*, 10 (4), 676–691.

Goldratt, E., 1990. *What is this thing called theory of constraints and how should it be implemented?* Croton-on-Hudson, NY: North River Press.

Goldratt, E., 1994. *It's not luck*. Croton-on-Hudson, NY: North River Press.

Goldratt, E.M. and Cox, J., 1984. *The Goal*. Croton-on-Hudson, NY: North River Press.

Gupta, A., 2010. Nimbus – An application to resolve conflicts using Goldratt's evaporating cloud. *Annual Meeting of American Junior Academy of Sciences*. San Diego.

Gupta, M. and Boyd, L., 2011. An excel-based dice game: an integrative learning activity in operations management. *International Journal of Operations & Production Management*, 31 (6), 608–630.

Gupta, M., Boyd, L.H., and Kuzmits, F., 2011. The evaporating cloud: A tool for resolving workplace conflicts. *International Journal of Conflict Management*, 22 (4), 394–412.

Hammer, M. and Champy, J., 1993. *Reengineering the corporation: A manifesto for business revolution*. New York: Harper Business.

Heineke, J.N. and Meile, L.C., 1995. *Games and exercises for operations management*. Upper Saddle River, New Jersey: Prentice-Hall.

Jackson, G.C., Stoltman, J.J., and Taylor, A., 1994. Moving beyond trade-offs. *International Journal of Physical Distribution & Logistics Management*, 24 (1), 4–10.

Loch, C.H. and Yaozhong, W., 2005. Behavioral operations management. *Foundations & Trends in Technology, Information & Operations Management*, 1 (3), 121–232.

Scheinkopf, L.J., 1999. *Thinking for a change: Putting the toc thinking processes to use*. Boca Raton: CRC.

Umble, E.J. and Umble, M., 2005. The production dice game: An active learning classroom exercise and spreadsheet simulation. *Operations Management Education Review*, 1 (1), 105–122.

Watson, K.J., Blackstone, J.H., and Gardiner, S.C., 2007. The evolution of a management philosophy: The theory of constraints. *Journal of Operations Management*, 25 (2), 387–402.