Using Conceptual System Dynamics Simulation Models to Initiate Exploration of and Discussion about Incident Reporting Systems

Finn Olav Sveen[1,2], Jose Mari Sarriegi[1], Jose J. Gonzalez[2], Siri Skaaland[3], Joakim von Brandis[3], Elin Brox[3], Geir Skjøtskift[3]

[1]Tecnun (University of Navarra), Manuel de Lardizábal 13, 20018 San Sebastián, Spain
[2]University of Agder, Faculty of Engineering and Science, Department of ICT, Security and Quality in Organizations, 4898 GRIMSTAD, Norway
[2]NISlab, Gjøvik University College, 2802 Gjøvik, Norway
[3]mnemonic as, Wergelandsveien 25, 0167 Oslo, Norway

Abstract

Ongoing research collaboration between Tecnun, University of Agder, Gjøvik University College and mnemonic AS (a Managed Security Services provider), investigates how to improve the operation of information security incident reporting systems. A large part of the research effort is collaborative workshops and a significant issue is how to engage the participants in an objective discussion. We have successfully employed small System Dynamics computer simulation models for this purpose. These models leave out many details and make a number of assumptions that are often wrong. However, that is precisely why they work so well. When experts are confronted with a “wrong” model of a system they know very well, they seem to have an urge to immediately correct the modeler, thus initiating discussion. Used correctly, these small conceptual models can “kick start” a collaborative modeling workshop, engaging the participants and immediately extracting useful information. This paper presents one such model and our experiences with using it.

Introduction

In an ongoing research project, MIRSA (Modeling Incident Reporting Systems and Awareness), universities and a private company aim to better understand and improve the operations of information security incident reporting systems (IRS). The university partners are Tecnun (University of Navarra) in Spain and the University of Agder and Gjøvik University College in Norway. The private company is mnemonic AS, one of Norway’s largest security companies. The project case is mnemonic’s internal IRS.

To achieve the goal of improving the operation of the IRS we build System Dynamics computer simulation models (Forrester 1958, 1961; Richardson and Pugh 1981; Sterman 2000) in small groups using a methodology called Group Model Building (Andersen and Richardson 1997; Richardson and Andersen 1995; Vennix 1996). In this approach the academic partners and the participants from mnemonic, collaboratively build models in a series of workshops lasting from half a day to a full day each. The participants from mnemonic have little or no background in System Dynamics (SD) and since no training would be given before the workshops, we needed an approach that would allow us to quickly introduce the crucial aspects of SD iconography in a short time. We wanted to spend as little time as possible training and as quickly as possible start fruitful discussion about the IRS.

The System Dynamics group at SUNY University at Albany has run projects using Group Model Building (GMB) for many years. To solve the above mentioned problem they use “concept models”. In the words of Richardson (Richardson 2006): *In the early exploratory days of group model building interventions at the University at Albany, we* *settled on the use of sequences of tiny models for this purpose, which we call "concept models." The term reflects the conceptual nature of these little models in two senses. The models introduce concepts, iconography, and points of view of the system dynamics approach. In addition, the models are designed to try to approach the group's own concepts of its problem in its systemic context.*

We decided to employ the approach of "concept models" in our own GMB project. What follows is a description of the "concept models" that we designed and our experiences in using it.

Conceptual Model of an IRS

The project so far has consisted of four workshops. The first workshop targeted a specific and bounded problem definition as well as identifying problem indicators and potential strategies to make those indicators behave ideally. As such, there was little need for the use of specific SD tools in the first workshop. We started building our SD model of mnemonic's IRS during the second workshop. For this purpose we built a concept model to be used early in the workshop. It was designed in line with the guidelines that Richardson presented (Richardson 2006). The model had several purposes:

1. To introduce SD notation.
2. To show the connection between the structure of the system and its behavior.
3. To motivate discussion about the group's problem.

Figure 1 shows the first SD symbols that the group from mnemonic was introduced to. The model was first hand drawn on a whiteboard and the symbols explained to the group. Richardson states that it is important to start with hand drawn symbols, but he does not say why. A possible explanation is that it may contribute to keeping attention on the symbols themselves and not the underlying mathematical expressions. The box in the middle is a stock or a level. An often used metaphor for it is the level of water in a bathtub. The "water" in our Open Incidents "bathtub" is the amount of open incidents that have not yet been handled by the incident reporting team. Into the "bathtub" flows reports of incidents and once they have been handled and analyzed, they flow out. We did not explain the mathematics to the group, but essentially it is an integration of the inflow – outflow over a time period.

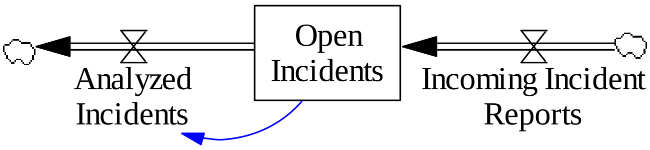

Figure 1 This first part of the conceptual model was hand drawn on a whiteboard before showing the computer model.

We then extended the hand-drawn model to what is shown in Figure 2 below. We did omit some of the constants from the hand drawn figure. However, we did reveal all of the variables and constants afterwards when the group was shown the computer model. Before switching to the computer we explained the basic feedback structure of the concept model. When a user reports, he or she expects to get feedback in a timely manner. If the response time between reporting and feedback becomes too long, it will effectively discourage reporting. The other way around is also true. A short response time encourages more reports. Hence, the capacity to analyze incidents has an impact on the motivation of the user.

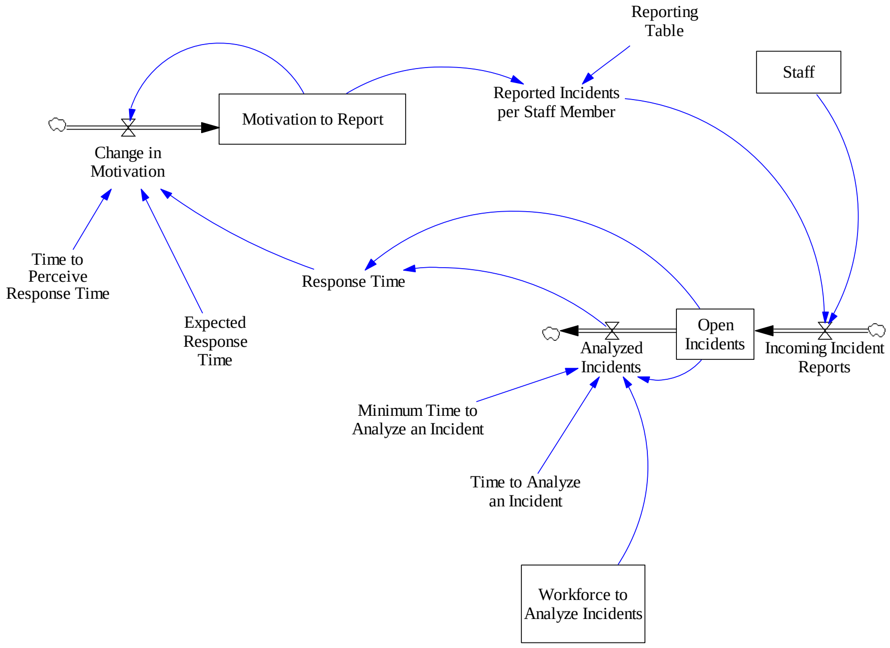
Reporting Table
Staff
Motivation to Report
Reported Incidents per Staff Member
Change in Motivation
Time to Perceive Response Time
Expected Response Time
Response Time
Open Incidents
Analyzed Incidents
Incoming Incident Reports
Minimum Time to Analyze an Incident
Time to Analyze an Incident
Workforce to Analyze Incidents

Figure 2 Concept Model Version 1. Motivation to report depends on feedback to user.

We then switched to the computer model, showed the group that it was essentially the same as the hand-drawn model and simulated it. The result of the simulation is shown in Figure 3. In the first five weeks of the simulation all incoming incidents are analyzed and handled with a minimum delay. In week 5 the number of incoming reports exceeds the capacity of the incident handling team to analyze them, as shown by the abrupt stop in the rise of Analyzed Incidents. Incoming Incident Reports keeps rising almost until week 10 as it takes some time before staff notices the increasing Response Time. Once the staff notices that timely feedback starts to become less than timely, their motivation to report incidents start to fall and with it the incoming incident reports. Eventually, incoming incident reports stabilize for a while, somewhat below the capacity of the incident handling team. Towards the end of the 100 weeks that are simulated, the motivation to report increases slightly as the incident handling team works off the backlog of reported incidents.

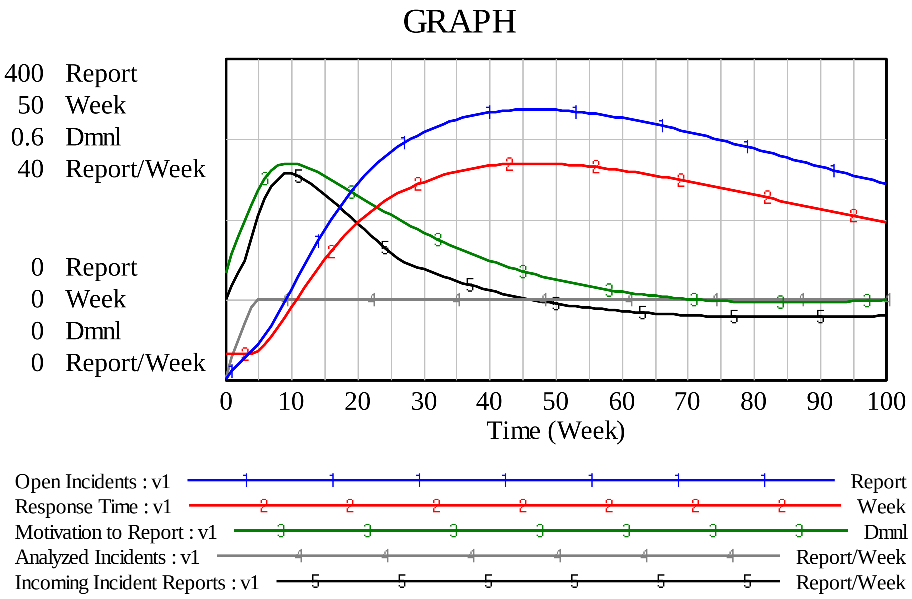

Figure 3 Behavior of Concept Model Version 1.

The more the users reported, the more they were being discouraged from reporting until in the end some balance was found. We explained to the group that we wanted to attempt to control the response time through other means. Instead of discouraging the users from reporting we wanted to encourage them. We then opened version two of our concept model. We did not draw this model by hand, as it would cause delay through switching on and off the projector.

Version two of the concept model includes structure that adds more workforce to analyze incidents when the response time increases. The version two model is shown in Figure 4.

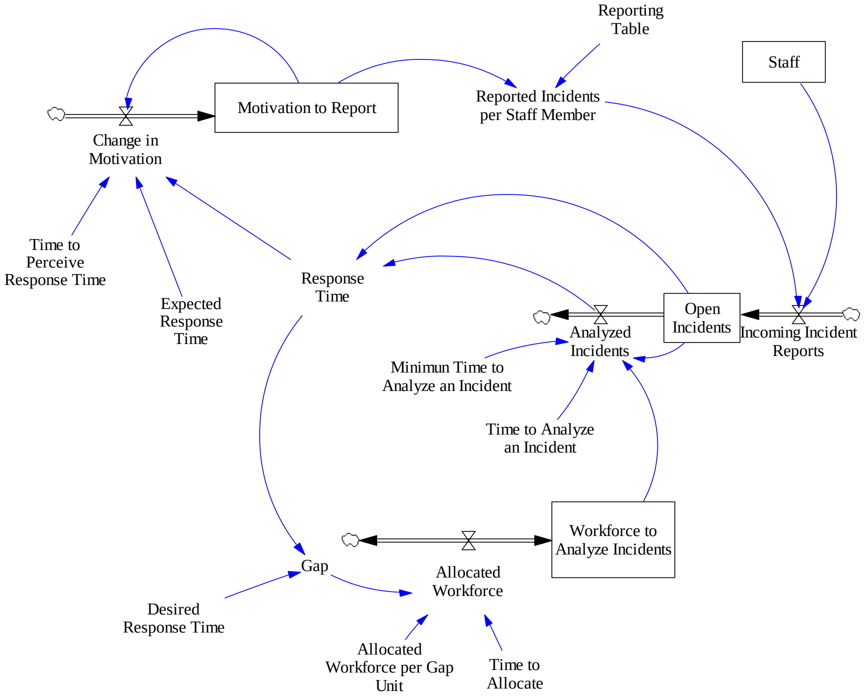
Reporting Table
Staff
Motivation to Report
Reported Incidents per Staff Member
Change in Motivation
Time to Perceive Response Time
Expected Response Time
Response Time
Open Incidents
Analyzed Incidents
Incoming Incident Reports
Minimun Time to Analyze an Incident
Time to Analyze an Incident
Workforce to Analyze Incidents
Gap
Allocated Workforce
Desired Response Time
Allocated Workforce per Gap Unit
Time to Allocate

Figure 4 Concept Model Version 2. Extra workforce is added to keep response time down.

After explaining the extension of the model, we simulated it in front of the group. The behavior is shown in Figure 5. This time the behavior is much more complex, but does retain some of the basic features of the previous model. As long as the response time is low, the motivation to report increases. When the response time increases, the motivation decreases, and also the number of reports. However, after a while management perceives the increasing response time and adds more workforce over time, driving the response time down. This reverses the users' falling motivation to report, causing more incident reports. Soon the new incident handling capacity is exceeded and response time falls yet again and with it motivation. Management is then forced to again add more workforce.

This behavior keeps repeating. In this case the strategy of throwing more resources at the problem clearly does not work in the long run.

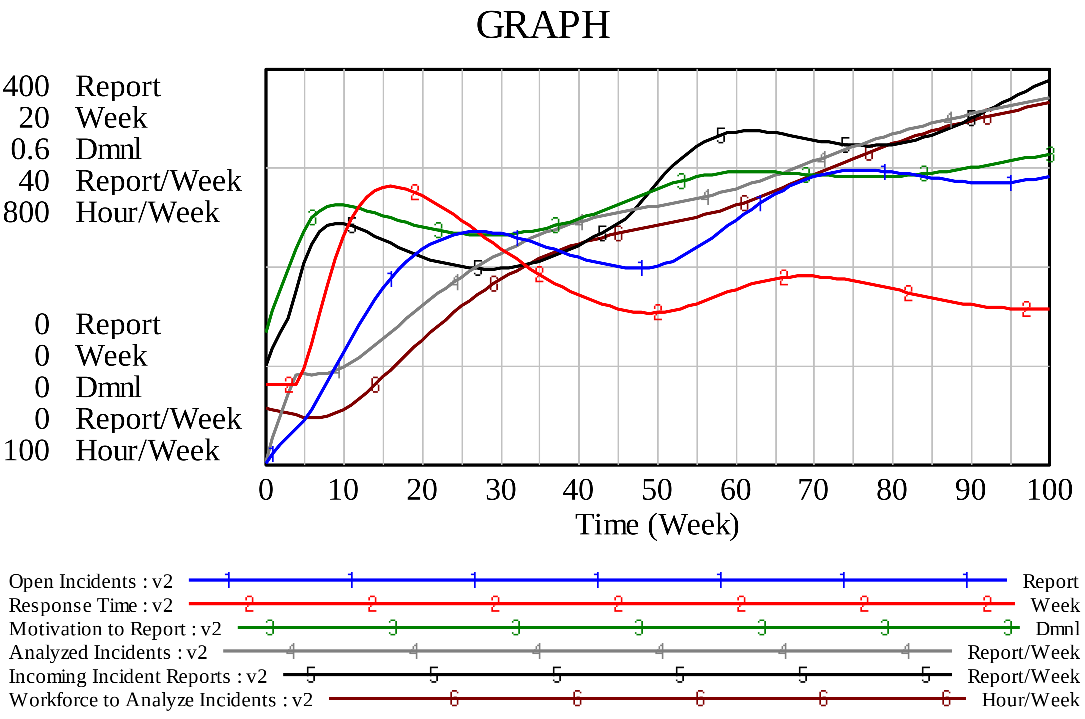

Figure 5 Behavior of Concept Model Version 2

Reactions to the Concept Model and Discussion

The concept models worked well in three ways: 1) It adequately demonstrated System Dynamics iconography. 2) It demonstrated the connection between system structure and behavior. 3) It was useful in starting a conversation about IRS, because it appeared as obviously wrong to the workshop participants.

We will here focus on the third point. When we started discussing the behavior and structure of the concept model some glaring shortcomings were immediately pointed out by the workshop participants. First of all, the model's only driver for user motivation to report is the incident response time. It is quite obvious that other factors also impact user motivation. These factors can be, e.g., management focus on reporting and the quality of feedback to the user and not just the time it takes to give feedback. The origins of security incidents were not part of the model either. The model simply assumed that users would report a number of incidents based on their motivation, the model did not have any connection between actual risk and reported incidents. Another observation was that most of the strategies that could be used to affect the behavior of the IRS were not included in the model, these were strategies that had been discussed in a previous workshop and their omission stood out. The strategy that was included, to throw more resources at the problem, blatantly ignored financial restrictions.

All of the above were pointed out by the mnemonic team. A lively discussion ensued, providing us with a great starting point for the next exercise of the day. This exercise was to elaborate on system resources that had been previously identified last workshop. The task was to identify the inflows and outflows that changed resources. In other words, what are the change mechanisms for, e.g., user motivation to report incidents, financial resources for security, incident handling knowledge, etc.?

We had achieved our goal of kick starting the discussion. We believe that this was because the model appeared as wrong to the participants, but as right enough that they could accept it as a representation (although a bad one) of the system they knew very well. In essence the model cried out to be fixed. Richardson (2006) gives an example of a concept model that did not adhere to the above mentioned guidelines. In his words:

Certainly, from the point of view of a professional modeler, the assessment model was trivially simple – linear, with exogenous time series for scenario parameters, essentially open loop, no rich feedback structure, no compensating feedback for policy initiatives, and so on. But the model looked right and behaved right to the participants, particularly after good parameter estimates were introduced. Unlike our usual concept models, it did not have rather glaring simplifications or inaccuracies. It did not cry out to be fixed. It did not provide the drive toward rich give-and-take conversation among the participants, facilitated by the modeling team, trying to get a systems view of the tough assessment problem. In fact, it may have looked like it "solved" the problems.

A model that looks like it solves the problem is certainly not good when the idea is to start a conversation. It will have the opposite effect of stifling it instead.

Compared to Richardson's ideal examples, our concept model does have some shortcomings. It has more mathematical detail than strictly necessary. The goal is not to create a model that adheres completely to good modeling practice and mathematical convention, rather it is to start a good conversation (Richardson 2006). The concept model is only a starting point for further model development, thus it does not need to be formally correct in all aspects. However, we did not notice negative effects with regards to this issue. The workshop participants never asked to see the mathematical equations behind the model, and as such we did not have to explain the mathematics of the model. We also did not follow entirely Richardson's recommendations with regards to model presentation (Richardson 2006). When we presented the second part of the concept model we did so using only a computer and projector. Richardson recommends that the model should always be drawn by hand first. We did not notice any negative effects of using only computer for the second part of the presentation. First, we did present the first part of the concept model by drawing it on the whiteboard, which might have mitigated a bit. Second, our audience was all computer savvy people, professionals working with information security; hence they were well acquainted with working on computers. It might be that this aspect would have been more important if we were working with a less tech savvy group.

In summary, our experience with the concept model presented in this paper has been very good and matches the experiences described by Richardson. Our case adds to the growing number of GMB projects who have been able to use concept models to good effect.

References

Andersen, David F., and George P. Richardson. 1997. Scripts for Group Model Building. *System Dynamics Review* 13 (2):107-129.

Forrester, Jay Wright. 1958. Industrial Dynamics: A Major Breakthrough for Decision Makers. *Harvard Business Review* 26 (4):37-66.

———. 1961. *Industrial Dynamics*. Cambridge MA: Productivity Press.

Richardson, George. 2006. Concept Models. Paper read at The 24th International Conference of the System Dynamics Society, July 23-27, at Nijmegen, The Netherlands.

Richardson, George P., and David F. Andersen. 1995. Teamwork in Group Model Building. *System Dynamics Review* 11 (2):113-137.

Richardson, George P., and Alexander L. Pugh, III. 1981. *Introduction to System Dynamics Modeling with DYNAMO*. Cambridge MA: Productivity Press.

Sterman, John D. 2000. *Business Dynamics: Systems Thinking and Modeling for a Complex World*. Boston: Irwin/McGraw-Hill.

Vennix, Jac A. M. 1996. *Group Model Building: Facilitating team learning using System Dynamics*. Chichester, England: John Wiley and Sons.