

Quantitative System Reliability Approach for Optimizing IT Security Costs in an AI Environment

Geert Alberghs¹, Pavel Grigorenko², and Jyri Kivimaa³

¹Ecole Supérieure d'Informatique Electronique Automatique, Paris, France, ✉ geert.alberghs@mil.be

²Institute of Cybernetics, Tallinn University of Technology, Tallinn, Estonia, ✉ pavelg@cs.ioc.ee

³Cooperative Cyber Defence, Centre of Excellence, Tallinn, Estonia, ✉ jyri.kivimaa@ccdcoe.org

Abstract

The Graded Security Model (GSM) addresses the IT Security cost optimization, problem by trying to find an answer to the following question: "For a certain budget level, in which IT security measures should be invested to achieve the highest possible overall security level?" This paper describes how reliability engineering can be applied to solve the GSM optimization problem. The organization's IT security measures are represented in a reliability block diagram, which in turn can be translated to an undirected graph. The total reliability of the diagram can be calculated after the identification of Minimal Cut Sets (MCSs). Cellular Automata (CA) are combined with Monte Carlo (MC) sampling to allow the identification of all MCSs. This approach allows the replacement of every possible user provided diagram by a series structure of parallel components, for which the total reliability can always be calculated. Additionally, this new model allows the calculation of cut set criticalities and component Fussell-Vesely (FV) importance values. All implementations have been realized with the Artificial Intelligence (AI) platform CoCoViLa.

1 Introduction

1.1 IT Security Investment Optimization

Information security has turned out to be a critical business component. The success of an organization is closely related to its ability to appropriately manage risks. That is why Cost-effectiveness analysis¹ software for security investments is now becoming an absolutely indispensable decision support tool.

Over the past few decades several models and frameworks have been suggested to help management with the selection of appropriate security measures. These models can be categorized into three main research areas.

The first type of models, the *think like an attacker* models ([4]), is the most intuitive. Sequential or tree analysis techniques are used to identify possible hacker actions. Security measure selection is based on incident likelihoods, cost-benefit criteria, pruning of duplicate security measures in the attack tree, etc. The main drawback of these models is that selection of security measures is only considered during the production phase and is not embedded in the Software Development LifeCycle (SDLC).

The problem that arises with the second type of models known as SDLC models ([5]), is the definition of security goals like Confidentiality, Integrity and Availability as functional requirements. This is why in some SDLC models ([9]) only best practices are implemented. The drawback here is that the commonly identified best practices might not be the optimal solution for a particular organization.

This paper is situated in the third research area: *Economics of Investments in Information Security*. In this field metrics as Return On Security Investments, Cost-Benefit analysis, Net Present Value and Internal Rate of Return are used to select the correct security measures. There are two subgroups of economic models: general economic models ([6]) which describe information security investment trends and laws,

¹Cost-Effectiveness analysis is distinct from cost-benefit analysis, which assigns a monetary value to the measure of effect.

and models that use economical measurements to identify, select and optimize security measures for a particular organization ([3]).

Our research is part of the second subgroup and uses Cost-Effectiveness analysis as a metric. Case studies have been performed based on Estonian SEB Bank and SwedBank expert data.

1.2 The Graded Security Model (GSM)

Selection of the right security measures appears to be a complex problem, because multiple objectives need to be achieved at the same time: Organisations need to:

1. attain their security goals,
2. with maximum efficiency and
3. at minimum cost².

The security goals to reach can be confidentiality, integrity and availability. Other security goals can be added according to specific organizational needs. (e.g. non-repudiation, authentication)

A major obstacle for finding a conclusive answer for the cost-effectiveness optimization issue in IT security is the lack of reliable metrics. In our Graded Security Model (GSM) the metrics of the NISPOM 2006 approach [17] are used to express the relations between security goals and security measure groups, where each security measure group i can be implemented at different levels l_i . As in [2] each level has additionally been characterized by its maintenance cost m_{i,l_i} , its investment cost i_{i,l_i} and its efficiency e_{i,l_i} . The efficiency levels are expressed as probabilities and indicate how confident³ we are that our security measure group implemented at a certain level, will not be the underlying reason of any security incident.

1.3 The Graded Security Expert System (GSES)

Based on the GSM described in Section 1.2 a cost-effectiveness analysis tool for IT security investments, the Graded Security Expert System (GSES), has been developed with the Artificial Intelligence (AI) software CoCoViLa [1, 7]. The GSES aims to maximize the overall system effectiveness E while staying within the available budget b , and this for a certain range of budget levels.

Now let the protection profile $p = (l_1, \dots, l_i, \dots, l_n)$ be the tuple representing a security level l_i for each security measure group between 1 and n .

The overall cost functions, Investment Cost $I(p)$, Maintenance Cost $M(p)$ and Total Cost $C(p)$ can then easily be written as follows:

$$I(p) = \sum_{i=1}^n i_{i,l_i} \quad (1)$$

$$M(p) = \sum_{i=1}^n m_{i,l_i} \quad (2)$$

$$C(p) = I(p) + M(p) \quad \text{where of course} \quad C(p) \leq b \quad (3)$$

1.3.1 The weighted average approach

The first versions of the GSES software [11, 13, 14, 15] used a weighted average for determining the overall efficiency $E(p)$. w_i represents the weight of security measure group i .

²Losses considerations have been omitted for reasons of clarity, but are definitely included in our cost-effectiveness analysis model and - tool

³The notion confidence can be considered as the exact opposite of the term likelihood used in risk management: $Likelihood = 1 - Confidence$

$$E(p) = \sum_{i=1}^n w_i e_{i,l_i} \quad \text{with} \quad \sum_{i=1}^n w_i = 1 \quad (4)$$

This method has several drawbacks. Users have difficulties assigning correct weights to each security measure group and since weights are constants there is no possibility to define dependencies between effectiveness values of security measure groups or to include the influence of the security goals.

1.3.2 The measure group relationship diagram

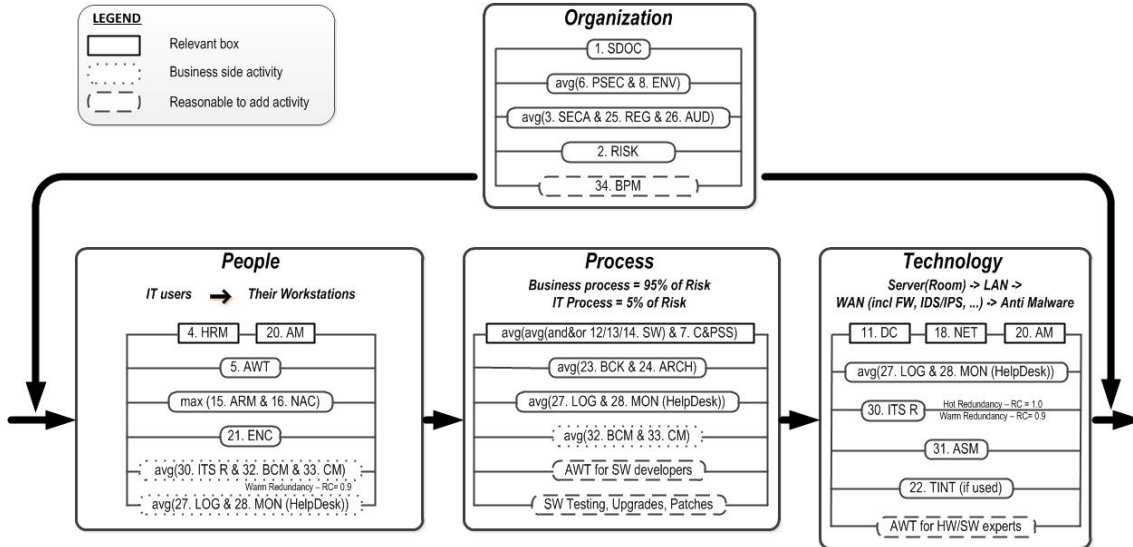


Figure 1: Example of a GSM measure group relationship diagram

To cope with the drawbacks mentioned in Section 1.3.1 the redundancy coefficient R_c has been introduced in [12]’s GSES version⁴ to represent inter-component dependencies in a measure group relationship diagram. The values of this coefficient are generally easier to estimate than the weights of (4).

R_c represents the dependency between a so-called relevant measure group and his supporting measure group. When establishing the structure of a system, it seems reasonable to be able to reduce the system to the components that play a direct role for the functioning ability of the system. The components we are left with are called relevant components. To avoid the usage of the term “irrelevant”, the components that are not relevant are called “supporting”. Supporting measure groups are always drawn parallel to the relevant measure groups they influence, the latter being outlined in red in Figure 1.

Good examples of supporting measure groups are “logging” and “monitoring”. They improve, for instance, an organization’s capability to detect hardware errors. The efficiency of the security measure group “redundant hardware” would thus clearly be influenced by changes in the implementation level(s) of “logging” and “monitoring”.

As for the diagram based calculations:

- A series configuration is always less efficient than its weakest component:

$$E(p) = \prod_{1 \leq i \leq n} e_{i,l_i} \quad (5)$$

⁴Although the new approach hasn’t been explicitly mentioned in this paper

- A parallel configuration is always more efficient than its strongest component:

$$E(p) = 1 - \prod_{1 \leq i \leq n} (1 - e_{i,l_i}) \tag{6}$$

- Finally redundant connections are calculated as follows:

$$E(p) = 1 - (1 - E_r) \prod_{1 \leq i \leq n} (1 - R_{c_i} e_{i,l_i}) \tag{7}$$

In (7) r represents the relevant measure group, with measure groups 1 through n supporting it and $\forall R_{c_i} \in [0, 1], 1 \leq i \leq n$. If $R_{c_i} < 0.1$ the influence of measure group i is probably too small to invest in it, and if $R_{c_i} = 1$ measure group i is said to be fully redundant.

Now, using (5), (6) and (7) the serial and parallel subsystems of the diagram can be recursively replaced by their single equivalent components until the overall efficiency is found.

The idea behind the relevant measure groups in the measure group relationship diagram is that if one of them fails ($e_{i,l_i} = 0$) the entire system should fail ($E(p) = 0$). This means that in this model relevant measure groups cannot be placed in parallel. Another problem is that relations between measure groups can *only* be serial or parallel: bridge-, star- and other topologies are not possible.

1.4 Information Security Models

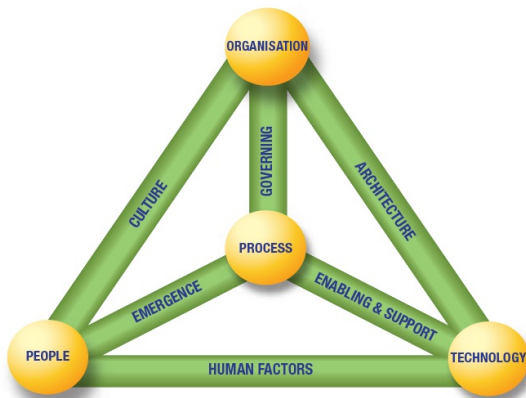


Figure 2: The Business Model For Information Security

ISACA mentions in [8] that until January 2009 there was no official holistic or dynamic model for security responsible to use as a guidance for managing IT security risks. There are many standards and frameworks to address specific needs, but no overarching model that could exist in any organization regardless of geographic location, industry size, regulation or existing protocol. In fact, the answer ISACA sought is exactly what is needed in the GSM to model security efficiency. Their solution is to represent an organization by using 4 elements and 6 dynamic interconnections as shown in Figure 2 and by assigning all organization's security measure groups to the correct elements and interconnections. Each security measure group can be present in more than one element and/or interconnection and

depending on its location it can be more or less efficient.

ISACA's Business Model for Information Security (BMIS) and the measure group relationship diagram suggest the usefulness of a graph structure for representing the security posture of an organization.

1.5 Improving the model

The main idea behind this paper is that the GSM should become a holistic model as the BMIS, able to represent all types of organizations. It cannot be subject to the limitations mentioned in 1.3.2. A solid mathematical background will be added. The efficiency levels, previously expressed as roughly estimated weights and confidence values, will be made more quantifiable. It will also be possible to prioritize among relevant security measure groups by using Fussell-Vesely importance values. Finally the Minimal Cut Sets (MCSs) concept will allow us to look at an organization through the attacker's eyes: A MCS actually is the smallest set of IT Security controls which, when disabled, prohibits an organization to reach its security goals.

2 Graph structure

An undirected graph (Figure 3) is used for modeling the security efficiency.⁵ The relevant measure groups are the edges of the graph connecting the circular nodes. The nodes are considered being fully reliable (efficiency of 1).

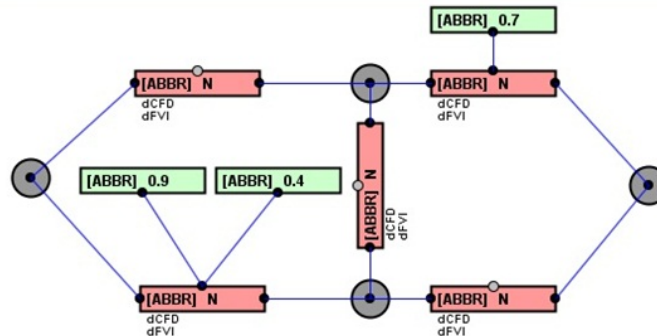
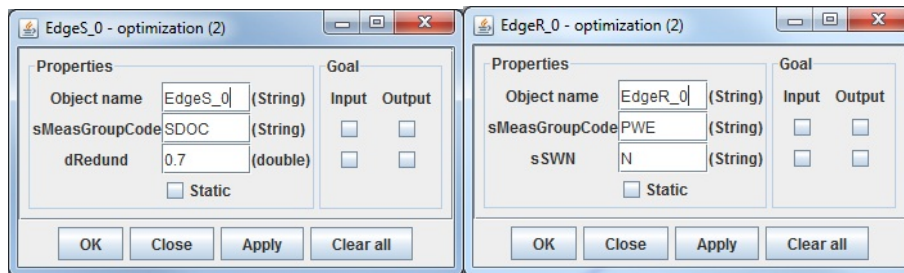


Figure 3: Graph structure used as a test-case in CoCoViLa

The boxes, connected to the relevant measuregroups which are not part of the graph structure as such, represent the supporting security measure groups. The properties of both types of security measure groups are shown in Figure 4 and explained in Table 2.



(a) Supporting Edge

(b) relevant Edge

Figure 4: Edge Properties in CoCoViLa

To reduce Figure 3 to a real graph structure, (7) is used between relevant and supporting security measure groups.

Variable	EdgeS	EdgeR	Description
String ObjectName	X	X	Name of the object instance
String sMeasGroupCode	X	X	Allows access to the cost and efficiency data of the referenced security measure group
String sSWN		X	Allows (e.g. ±10%) variations on efficiency values as suggested by the BMIS Strong, Weak, Neutral attributes
double dRedund	X		Redundancy coefficient

⁵Please note that this graph is a simplified representation, used only for testing purposes. It doesn't represent any existing organization.

3 System Reliability Approach

3.1 Introduction

Threats exploit vulnerabilities and manifest themselves through a certain impact on the organization. Impacts can be measured rather easily, threats and vulnerabilities unfortunately not. All information about the measure group is described with the probability density function $f(t)$ of its time to failure T . No explicit modeling of the threats and vulnerabilities is carried out. Reliability characteristics like *failure rate* and *Mean Time To Failure (MTTF)* are deduced directly from the probability density function $f(t)$. After several components (measure groups) are combined into a system (organization) a System Reliability Analysis can be performed.

By applying the ISO 8402 definition of reliability to our model, efficiency can be formulated as: *the ability of the security measure group to perform a required security function, under given threats and vulnerabilities and for a stated period of time.*

To verify if the measure group performs its required security function:

1. the security incidents need to be recorded in an incident management system
2. the causes of the incidents need to be identified. (i.e. find out which security measure group failed)
3. the *failure rate* of the involved measure group must be updated after each incident

The next section explains how the measure group efficiencies can be derived from these failure rates.⁶

3.2 Incident model

A well maintained and updated measure group can be considered as good as new during its entire useful lifetime, meaning that the failure rate λ is approximately a constant⁷. That is why the exponential distribution, the most commonly used life distribution in applied reliability analysis, can also be used in the GSES. Its main benefits are its mathematical simplicity and that it has often proved to lead to realistic lifetime models.

The definition of an exponential distribution is as follows:

$$f^i(t) = \begin{cases} \lambda_i e^{-\lambda_i t} & \text{for } t > 0 \quad \lambda_i > 0 \\ 0 & \text{otherwise} \end{cases} \quad (8)$$

where $f^i(t)$ represents the probability density function of the time to failure T for a certain measure group i . The cumulative distribution function then becomes:

$$F_T^i(t) = Prob(T \leq t) = \begin{cases} 1 - e^{-\lambda_i t} & \text{for } t > 0 \quad \lambda_i > 0 \\ 0 & \text{otherwise} \end{cases} \quad (9)$$

Its reliability or efficiency function can then be written as:

$$E_T^i(t) = Prob(T > t) = 1 - F_T^i(t) = e^{-\lambda_i t} \quad \text{for } t > 0 \quad (10)$$

with $MTTF = 1/\lambda_i$ and the failure rate function $z^i(t) = \lambda_i$

So this means that:

- A measure group in use is always considered as good as new
- Only one parameter $\lambda_i = \frac{\text{Number of Incidents for security measure group } i}{\text{Observation Time}}$ needs to be collected (or estimated by experts) for each measure group

⁶For more in-depth explanations about reliability engineering please read [16], which has been used as the mathematical basis for this section.

⁷Burn-in and wear-out periods are not considered here. Extra caution is always needed during implementation and retirement phases of security measures. In IT security particular attention should also be devoted to the fast technological evolutions.

Pseudo random generators always follow a uniform distribution $U(0, 1)$, but random incidents against our measure groups respecting an exponential distribution can be simulated by applying the probability integral transform. The probability integral transform says that if a variable T has a continuous distribution for which the cumulative distribution function is $F_T(t)$, then the random variable $Y = F_T(t)$ has a uniform distribution.

Applied to our exponential distribution one can easily obtain the following equation:

$$T = \frac{-1}{\lambda_i} \ln(1 - U) \quad \text{with} \quad F_U(u) = \begin{cases} 1 & \text{for } 0 \leq u \leq 1 \\ 0 & \text{otherwise} \end{cases} \quad (11)$$

So now we are able to randomly generate incidents for the measure groups by:

1. setting the observation time t
2. randomly generating a sample U from a uniform distribution $F_U(u)$
3. calculating T according to (11)
4. comparing T with t : if $T < t$ then an incident has occurred

Now the GSES is able to model the security efficiency and to generate random incidents for all measure groups separately. The next step is the definition of the overall efficiency.

3.3 Reliability Block Diagrams (RBDs)

A Reliability Block Diagram (RBD) is a success-oriented network describing the function⁸ of the system. It shows the logical connections between components needed to fulfill this specified system function.

In the GSM matters have been simplified by assuming that the components are non-repairable and that the order in which the incidents occur does not matter. When the systems are repairable and/or the order in which failures occur is important, the more complex Markov methods should be used. In Markov methods the different states of the system need to be defined and the probabilities of transition between states should be estimated. The former is difficult but feasible, the latter however would be an almost impossible task in our case.

A system composed of n components will be denoted a system of order n . The set of components is denoted by:

$$C = (1, 2, \dots, n)$$

For both the components and the system itself a distinction between a functioning and a failed state is made. The state of component i , $i = 1, 2, 3, \dots, n$, can then be described by the binary variable x_i , where

$$x_i = \begin{cases} 1 & \text{if component } i \text{ is functioning} \\ 0 & \text{if component } i \text{ is in failed state} \end{cases} \quad x = (x_1, x_2, \dots, x_n) \quad \text{is called the } \textit{state vector} \quad (12)$$

Similarly the state of the system can be described by a binary function

$$\phi(x) = \phi(x_1, x_2, \dots, x_n)$$

where

$$\phi(x) = \begin{cases} 1 & \text{if the system is functioning} \\ 0 & \text{if the system is in failed state} \end{cases} \quad (13)$$

and $\phi(x)$ is called the structure function of the system and can be written as:

$$\phi(x) = \begin{cases} \prod_{i=1}^n x_i & \text{for serial components} \\ 1 - \prod_{i=1}^n (1 - x_i) = \prod_{i=1}^n x_i & \text{for parallel components} \end{cases} \quad (14)$$

⁸In our case the system function would be: provide security to the organization

3.4 Minimal Cut Sets(MCSs)

A cut set κ is a set of components in C which by failing causes the system to fail. A cut set is said to be minimal if it cannot be reduced without losing its status as a cut set.

Now consider a saboteur who wants to bring the system in a failed state, with the least possible effort on his/her part. What the saboteur would need is a list of the MCSs of the system.

With the definition of MCSs in mind the structure function can be rewritten as:

$$\phi(x) = \prod_{j=1}^k \kappa_j(x) = \prod_{j=1}^k \prod_{i \in \kappa_j} x_i \quad (15)$$

Until now our model was deterministic in nature, but the state variables x_i of the n components should be looked at as random statistical variables $X_i(t)$ representing the statistical events of security incidents occurring. The state vector 12 and system structure function (15) should be adapted accordingly.

$$X(t) = (X_1(t), X_2(t), \dots, X_n(t)) \phi(X(t)) = \prod_{j=1}^k \prod_{i \in \kappa_j} X_i(t) \quad (16)$$

Because the distributions of the state variables $X_i(t)$ are known (9), the structure function of the complete system can be calculated by using the MCSs as shown in (16).

In our environment the saboteur would be called the attacker or the threat; the system is referred to as the organization and failures would be replaced by security incidents. The structure function represents the overall security efficiency of the organization.

The only remaining problem now is to find an algorithm which is able to find all cutsets in a given reliability diagram.

4 Minimal Cut Set search algorithm

4.1 Introduction

A methodology based on a combination of Cellular Automata (CA) and Monte Carlo (MC) sampling is used to identify the MCSs of our system reliability diagram. A ranking of the measure groups criticalities can be achieved through the calculation of their Fussell-Vesely importance values.⁹

A candidate cut set is generated by using the probability integral transform as explained in Section 3.2. CA will be used to decide if the generated set of failed edges is a cut set or not. And finally MC will allow us to determine the MCSs.

4.2 Cellular Automata(CA)

To verify if a connection between the source and the target still exists, after applying random failures to the edges of our graph, CA is used. Consider a graph containing n nodes with a source node S and a target node T . Each node i can be in 2 states: active ($s_i(t) = 1$) or passive ($s_i(t) = 0$) and each edge ij can be in 2 states: success ($e_{ij}(t) = 1$) or failure ($e_{ij}(t) = 0$). The transition rule which is used for our particular CA setup is very simple: a node may only be activated (1) if there is at least one active node in its neighborhood and (2) if the edge connecting it to this node has not failed. This can be formulated as follows:

$$s_i(t) = (s_p(t) \wedge e_{ip}(t)) \vee (s_q(t) \wedge e_{iq}(t)) \vee \dots \vee (s_r(t) \wedge e_{ir}(t)) \quad \text{with } p, q, \dots, r \in N_i \quad (17)$$

The neighborhood N_i of each node can be determined by using the adjacency and incidence tables representing the graph.

⁹A similar approach is proposed for the assessment of the unreliability of complex networks in [18]

The algorithm then goes as follows:

- | | |
|---|---|
| 1. step $t = 0$ | 5. update the node states according to rule 17 |
| 2. set all node states to passive: $\forall i : s_i(0) = 0$ | 6. if $s_T(t) = 1$ stop (a path has been found) |
| 3. activate the source node: $s_S(0) = 1$ | 7. else if $t < n - 1$ go to 4 |
| 4. step $t = t + 1$ | 8. else $s_T(t) = 0$ and no path has been found |

4.3 Monte Carlo(MC)

To determine which cut sets are minimal the following algorithm is used:

1. the candidate is compared to MCS of lower order already present in the archive of MCSs. If one of these cut sets is included in the sampled one, the counter associated to this cut set is incremented by one. Otherwise,
2. the candidate is compared to the cut sets of the same order in the archive to check if it is already present. If so the associated counter is incremented by one. Otherwise,
3. the candidate is added to the archive with its counter set to 1 and it is compared with higher order cut sets to verify if it is included in any of them, in which case they are deleted and the associated counter is added to the counter of the newly found.

Of course one can never be sure that the algorithm has been exhaustive in finding the MCSs, but even with a relatively low number of trials the most probable MCSs will be found.

If a MCS is not found, it is highly probable that it contains measure groups with high efficiencies.¹⁰

Let M_A be the MCS with the smallest probability to be found during the Monte Carlo sampling. This means that M_A has the highest efficiency of all MCSs. (Its value will be the closest to "1") And since the overall efficiency $E(p)$ is calculated as a series structure of all MCS we can say that M_A is the MCS that influences $E(p)$ the least.

So the MCS that have not been identified by the Monte Carlo algorithm are the ones with the smallest influence on the overall efficiency.

Additionally, for the optimization itself the exact calculation of $E(p)$ may not be required. One only needs to be able to compare different candidate solutions to each other and select the best.

4.4 Fussell-Vesely Importance values

We are not only able to identify the cut sets. The criticality of each edge can also be computed using the Fussell-Vesely importance measure. It is computed as the ratio between the number of occurred cut sets containing edge ij and the number of Monte Carlo trials performed.

The higher the Fussell-Vesely importance value, the more critical the edge is. This can be due to:

- its efficiency: A low efficiency value, implies higher probabilities of being selected as a failed edge in the graph;
- its location in the graph: Generally the closer the edges are to the Source and or Target node, the more important they get and;
- the number of downstream edges connected to it: Generally the more edges that are connected, the more critical the edges becomes.

A good understanding about these effects for each particular measure group within the organizational graph structure will be of great value to security managers, because it will enable them to correctly prioritize among IT security investments.

¹⁰Since each MCS is a set of parallel measure groups we can also say that its efficiency is always higher than the highest efficiency of its components.

5 CoCoViLa implementation

5.1 Introduction

CoCoViLa is an Artificial Intelligence software development platform. It synthesizes algorithms based on inputs from attribute declarations, bindings between attributes, attribute dependencies and goals using its declarative specification language. The realization of the dependencies are pure Java methods. More information about the tool can be found in [1, 7].

The CoCoViLa platform contains:

- a Class Editor for creating the domain-specific language, defining class properties and their visual representations;
- a Scheme Editor which allows users to:
 - visually specify computational problems by drawing objects/instantiating classes on schemes,
 - set values of object properties
 - define relations between object attributes,
 - make use of expert tables
- a synthesizer built into Scheme Editor for generating Java programs from schemes

5.2 The scheme

The scheme created in CoCoViLa is shown in Figure 5. Its components are:

- The security measure groups (vertically aligned purple boxes), containing the investment costs, maintenance costs and efficiencies for each level of implementation.
- the graph structure as explained in Section 2.
- the superclass (blue box), collecting all attribute values through the specification language’s alias mechanism and containing all references to the “hidden”¹¹ classes containing the actual reliability calculations.
- the optimizer (green box) collecting all the optimization results and containing the GSM cost and efficiency functions plus a reference to a “hidden” class with a slightly modified Evolutionary Algorithm. (the original algorithm is described in [10])
- the security class (red box) containing all information about the security goals together with the losses calculation. (not covered in this paper)
- a graph2D object allowing the representation of our optimization results in a 2D graph.

The bindings between the different components represent their equality. They allow the exchange of values between classes.

The superclass also allows the selection of different efficiency levels. One can choose between the usage of:

- the current efficiency levels of the measure groups. This way the overall efficiency and importance values reflect the situation of the organization as it is.
- the efficiency levels of the measure groups as required by the security goals. This way the overall efficiency and importance values reflect the situation of the organization as management wants it to be.
- an average efficiency value which is the same for all measure groups. This can be used when
 - no other data is available or
 - when only the influence of location and number of connections need to be investigated
 - when all MCSs need to be found, not only the most probable ones.

¹¹hidden in this context means: not visible for a normal user. More details can be found in the next section

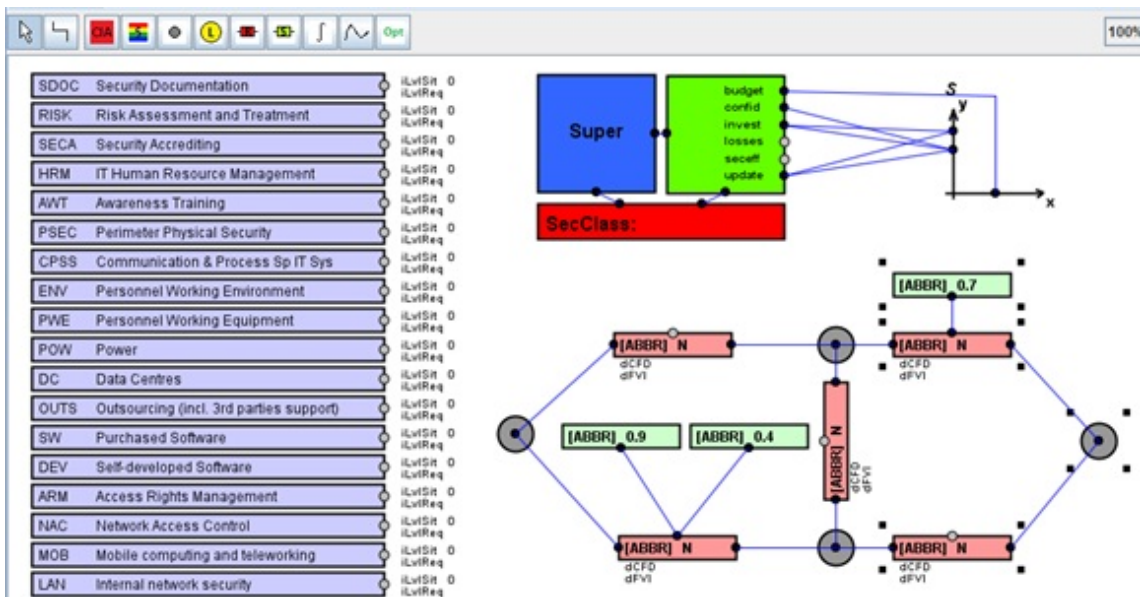


Figure 5: Representation of the optimization problem in a CoCoViLa scheme

6 Conclusion

A solid mathematical background has been added to the efficiency function $E(p)$ of the GSES. The definition of $E(p)$ evolved from a weighted average to a measure group relationship diagram and finally in this paper to a graph based reliability diagram with which the following benefits could be realized:

1. The efficiency of the measure groups is now quantifiable, unlike the previously defined weights and confidence levels. For this only the recording of the security incidents is needed.
2. CoCoViLa's user interface can be used to create the graph structure. No hardcoding of parameters is needed anymore.
3. The new approach is generic. A solution for all possible graphs can be found contrary to the measure group relationship diagram where parallel relevant measure groups, bridge- and star-topologies were not allowed.
4. The Fussell-Vesely importance values allow security managers to prioritize among IT security investments.
5. Threats and vulnerabilities don't need to be modeled, only incidents. Additionally MCSs reflect an attackers point of view with regard to the organization's security.
6. The influence of the security goals can be included by using the required measure group efficiencies as an input
7. It is justified by a well-founded mathematical theory

Recording security incidents and finding out which measure groups have failed remain non-trivial tasks however. Also for each measure group needs to be defined what would be considered as an incident. This definition will largely affect the measure group's failure rate. But if no statistical security incident information is available, the efficiencies can still be estimated by security experts as it was done before.

Important to know is that Monte Carlo sampling might not find all MCSs, but as stated in Section 4.3 this isn't absolutely necessary either.

The new version of the GSES should also be applied to an existing organization to verify if the predicted losses and efficiencies will correspond to the real losses and efficiency values.

References

- [1] CoCoViLa: Model-based software development platform.
- [2] Cyberprotect, version 1.1, July 1999.
- [3] J.L. Duffany. Optimal resource allocation for securing an enterprise information infrastructure. In Association for Computing Machinery, editor, *LANC7*, pages 35–42. Association for Computing Machinery, October 2007.
- [4] J.L. Duffany. Exploring security countermeasures along the attack sequence. In *ISA8*, pages 427–432. IEEE, April 2008.
- [5] I. Flechais, C. Mascolo, and M.A. Sasse. Integrating security and usability into the requirements and design proces. *International Journal of Electronic Security and Digital Forensics*, 1(1):12–26, 2007.
- [6] P.L. Gordon and P.M. Loeb. The economics of information security investments. *ACM Transactions on Information Security*, 5(4):438–457, 2002.
- [7] Pavel Grigorenko. *Higher Order Attribute Semantics of Flat Languages*. PhD thesis, Institute of Cybernetics at Tallinn University of Technology, Akadeemia Tee 21, 12618 Tallinn, November 2010.
- [8] ISACA. *The Business Model for Information Security*. ISACA, September 2010.
- [9] R.L. Jones and A. Rastogi. Secure coding: building security into the software development lifecycle. *Information Systems Security*, 13(5):29–39, 2004.
- [10] Toomas Kirt and Jyri Kivimaa. Optimizing IT security costs by evolutionary algorithms. In C.Czosseck and K. Podins, editors, *CCDCOE*, pages 145–160. Cooperative Cyber Defence Centre Of Excellence, June 2010.
- [11] Jyri Kivimaa. Applying a cost optimizing model for IT security. In Henrique Santos, editor, *ECIW*, pages 142–153. UK Reading, Academic Publishing Limited, July 2009.
- [12] Jyri Kivimaa and Toomas Kirt. Evolutionary algorithms for optimal selection of security measures. In Rain Ottis Demergis, editor, *ECIW*, pages 172–184. Readings, UK, Academic Publishing Limited, July 2011.
- [13] Jyri Kivimaa, Andres Ojamaa, and Enn Tyugu. Graded security expert system. In Roberto Setola and Stefan Geretshuber, editors, *CRITIS*, volume 5508 of *Lecture Notes in Computer Science*, pages 279–286. Springer, October 2008.
- [14] Jyri Kivimaa, Andres Ojamaa, and Enn Tyugu. Pareto-optimal situation analysis for selection of security measures. In *MILCOM*, pages 3224–3230, November 2008.
- [15] Jyri Kivimaa, Andres Ojamaa, and Enn Tyugu. Managing evolving security situations. In *MILCOM*, pages 1–7. Institute of Electrical and Electronics Engineers(IEEE), October 2009.
- [16] Marvin Rausand and Arnljot Hoyland. *System Reliability Theory, Models, Statistical Methods and Applications*. John Wiley & Sons, 2 edition, 2004.
- [17] U.S. Department of Defense, Defense Security Service. *National Industrial Security Program Operating Manual*, February 2006.
- [18] E. Zio, M. Librizzi, and G. Sansavini. Determining the minimal cut sets and Fussell-Vesely importance measures in binary networks by simulation. In Guedes Soares & Zio, editor, *ESREL*, volume 1, pages 723–729. Taylor & Francis Group, London, October 2008.

Institute of Cybernetics at Tallinn University of Technology

**12th Symposium on Programming Languages and
Software Tools**

SPLST'11

Tallinn, Estonia, 5–7 October 2011

Proceedings

TUT
PRESS

Tallinn © 2011

12th Symposium on Programming Languages and Software Tools
SPLST'11
Tallinn, Estonia, 5–7 October 2011
Proceedings

Edited by Jaan Penjam
Cover page design by Aive Kalmus

Institute of Cybernetics at Tallinn University of Technology
Akadeemia tee 21, EE-12618 Tallinn, Estonia
<http://www.ioc.ee/>

TUT Press
Akadeemia tee 1, EE-12618 Tallinn, Estonia

Sponsored by Estonian Centre of Excellence in Computer Science, EXCS
(funded mainly by the European Regional Development Fund)



Euroopa Liit
Euroopa
Regionaalarengu Fond



Eesti tuleviku heaks

ISBN 978-9949-23-178-2

Copyrights: the editor and authors, 2011

Institute of Cybernetics at Tallinn University of Technology

**12th Symposium on Programming Languages and
Software Tools**

SPLST'11

Tallinn, Estonia, 5–7 October 2011

Proceedings

TUT
PRESS

Tallinn © 2011

12th Symposium on Programming Languages and Software Tools
SPLST'11
Tallinn, Estonia, 5–7 October 2011
Proceedings

Edited by Jaan Penjam
Cover page design by Aive Kalmus

Institute of Cybernetics at Tallinn University of Technology
Akadeemia tee 21, EE-12618 Tallinn, Estonia
<http://www.ioc.ee/>

TUT Press
Akadeemia tee 1, EE-12618 Tallinn, Estonia

Sponsored by Estonian Centre of Excellence in Computer Science, EXCS
(funded mainly by the European Regional Development Fund)



ISBN 978-9949-23-178-2

Copyrights: the editor and authors, 2011



12th Symposium on
Programming Languages and Software Tools

Proceedings



TTÜ KÜBERNEETIKA INSTITUUT
Institute of Cybernetics at TUT