Pareto-Optimal Architecture according to Assurance Indicators*

Frank Innerhofer-Oberperfler
University of Innsbruck, Austria
frank.innerhofer-oberperfler@uibk.ac.at

Fabio Massacci, Artsiom Yautsiukhin†
Univesity of Trento, Italy
{Fabio.Massacci,evtiukhi}@dit.unitn.it

Abstract

In this paper we present an approach and algorithm for selecting the "best" secure architecture for supporting a business process according to a variety of assurance indicators. The key difficulty is to select an architectural design in presence of multiple indicators that might offer alternative notions of minimality. Therefore we must use the notion of Pareto optimality in order to select alternatives that are not dominated by others.

1 Introduction

The financial and management scandals of the last few years have spurred legislators and regulators of different countries to take initiative to protect customers and shareholders. While legislators do not mandate how business should be conducted, they require that companies should show that "they are in control of their business" and that such control is correctly reported to interested stakeholders [7].

Since most business processes are automated or rely heavily on an IT infrastructure such business controls are in practice very often controls embedded in information systems or controls supported by IT. Companies are therefore required to show that they have in place the necessary IT controls and that they have ways to assess their control system and control levels.

According to COBIT [7, pag.9]:

> Enterprises need an objective measure of where they are and where improvement is required, and they need to implement a management tool kit to monitor this improvement. Figure1 shows some traditional questions and the management information tools used to find the responses, but these dashboards need indicators, scorecards need measures and benchmarking needs a scale for comparison.

Obviously the ideal notion of indicator for a manager is the dashboard with red, green and yellow. Unfortunately life is more complex and often we must make decisions considering a variety of indicators. We must judge how the compliance goals of a company are reached by considering multiple and possibly conflicting indicators.

Management guidelines such as CoBIT describe how to set control goals (for security and assurance) and provide guidance in the definition of indicators. But such guidelines do not specify how one can use indicators of individual control goals to assess the overall enterprise architecture. In our previous paper [5]

*This work was partly supported by the EU-IST-FP6-IP-SERENITY and IST-FP6-IP-SENSORIA projects.

†Contacting Author: evtiukhi@disi.unitn.it

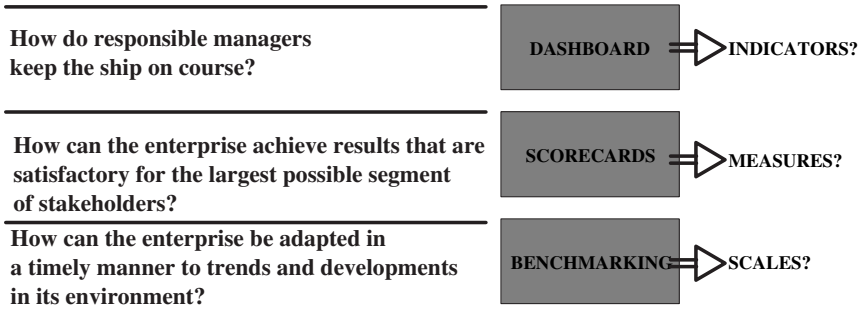

Figure 1: Management Information - From CoBIT

we have outlined how a security analyst can evaluate the impacts of security breaches on business objectives of an enterprise. In [16, 15] we provided an approach for aggregating security indicators for a business process (BP) and for selecting the most secure process model. However, both methods have a limitation: the use of a *single* indicator.

1.1 Contribution of this paper

In this paper we combine both models and consider the case of an evaluation against *multiple-indicators*, which allows an analyst to consider several security criteria at once. We also provide an algorithm for the aggregation of indicators and for choosing the "best" security architecture. In this case however we need to have a criterion for optimality that does account for possibly conflicting criteria. A practical example could be the consideration of indicators for compliance with European privacy directives and US Surveillance laws, which might be conflicting and negatively related.

To this extent we have have chosen the notion of Pareto-optimality: all indicators of a best solution should not be dominated by another solution. This means that we no longer have "the" best solution, but only "a" best solution. All other best solutions improve one indicator at the expense of some other indicator. The final tradeoff between the incomparable alternatives might then be evaluated using different business criteria. In practice an EU company doing business in the US might choose a different solution than a US company doing business in the EU. Still we want to be sure that there are no solutions which can better accommodate both compliance indicators.

This paper is structured as follows: By using an example case (Section 2) we outline and present the underlying Enterprise Model (Section 3). In Section 4 a mathematical model based on the underlying models is presented and followed by an algorithm for assessing the best available solution from a security point of view (Section 5). Finally, we outline some related work in Section 6 and give a conclusion and an outlook.

2 Running Example

Example 1 *In the paper we use our usual loan processing example. Consider a bank holding company which outsources the concrete loan processing to a semi-independent subsidiary. In order to provide the service, e.g., the loan originating process, the subsidiary needs to design the BP, find outsourcing partners and allocate the information systems (applications, servers, databases, etc.) that support the BP. There are a number of design alternatives to choose from: various parts of the BP may be fulfilled in different*

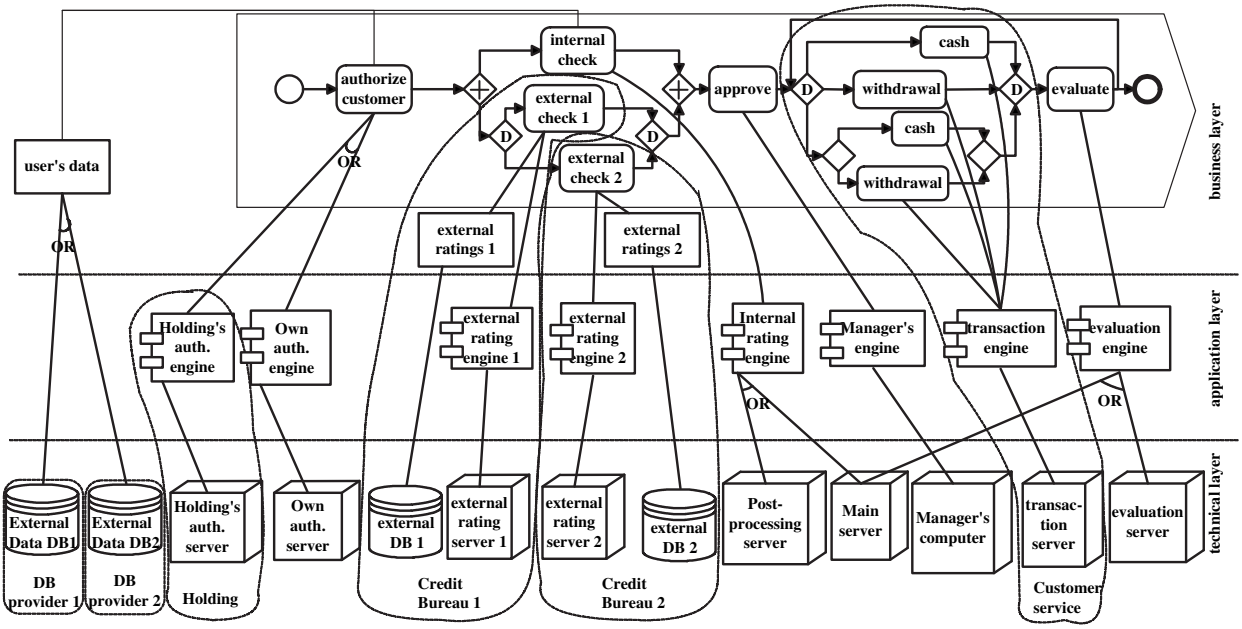

Figure 2: Architecture for loan origination business process.

ways, several outsourcing partners are available which offer the same functionality but provide different protection and trust levels, and last but not least applications and servers are also subjects for various design alternatives. Before providing the service the subsidiary must consider these different alternatives and choose the one which best fits the business needs.

The holding company is aware of a huge number of risks and losses caused by frauds[1]*. Therefore the holding wants to be sure that it is well protected against this type of losses. That is why one of the main criteria for choosing one of the design alternatives for the subsidiary is a low risk of possible frauds.*

For the BP modelling we use BPMN (Business Process Management Notation) [20] which is a widely used notation. We added just a small modification: the *design choice* gateway (diamond with letter D inside) to denote an alternative sequence of actions. After the selection only one path (alternative) should be left, i.e., all design choice gateways disappear and we receive a standard conformant BPMN diagram.

Example 2 *For our running example the following – reduced and schematic – model has been determined (Figure 2, upper right corner).*

In the Figure 2 one may see four main parts of the process: AUTHORIZE CLIENT, CHECK TRUSTWORTHINESS RATING, APPROVE *(and finalize) the loan and* REPAYMENT. CHECK TRUSTWORTHINESS RATING *consists of two activities fulfilled in parallel:* INTERNAL CHECK, *done by the subsidiary itself, and* EXTERNAL CHECK, *outsourced to one of two available Credit Bureaus.* REPAYMENT *is an iterative subprocess. Every month a customer pays the agreed amount and the transaction is* EVALUATED. *There are three possible ways of payment: payment by* CASH, *automatical* WITHDRAWAL *from the client's account or the possibility for the client to choose how to pay.*

[1] in average organizations loose 5% of annual revenue to frauds and abuses and that for banking companies, in particular, median losses are 258 000$ per company [1]

3 Enterprise Model

For our assessment we need a business oriented model of an enterprise which will help us to separate the different sources of threats on the one hand and provides a way to aggregate the data received from the sources on the other one. Typical sources for data in security systems are reports about security incidents derived from logs. Before using these data we should filter only the events which are relevant for our target of evaluation, i.e., the chosen BP.

In our previous paper we presented the security Enterprise Model based on ideas from [4, 12]. This model allowed us to aggregate security indicators, but did not provide a possibility to make a decision about the best enterprise architecture if several alternatives were possible. Also this model considered filtering of the events relevant for the chosen BP implicitly while in this work we make use the structure of the BP to make the filtering more explicit. In this paper we just outline the main concepts of the enterprise modelling which are relevant for the current paper.

The enterprise meta-model contains three layers (Figure 3):

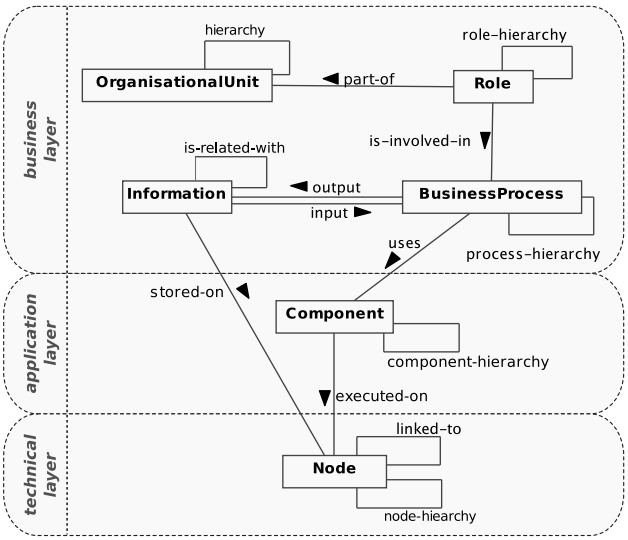

Figure 3: Enterprise meta-model

- **Business Layer.** The business layer contains business oriented artifacts. *Business process* is a predefined sequence of activities leading to the accomplishment of a goal. *Information objects* depict the information which is processed by business processes. The model contains also organizational units and roles, which are not relevant for this paper since we focus only on incidents impacting BP.

- **Application Layer.** On the application layer we have Components. *Components* are the information systems and applications used by some BP (by one of its activities).

- **Technical Layer.** The technical layer contains nodes. *Node* is a software and hardware set which provide required services for components to operate or information objects to be stored. For example, node can be a server with installed Windows Server 2003 and Oracle database.

Using this meta-model we can create an Enterprise Model which can be seen as a tree starting with a chosen BP and containing all elements connected to the BP through a functional dependency relationship.

This Enterprise Model presents in detail all artifacts affecting the chosen BP. The model also explicitly shows the sources of information about security violations related to the artifact.

Example 3 *There are several architectural design alternatives which must be considered in the example. The first one is two possible data bases where* USER'S DATA *can be stored. Another decision to be made is whether to use the holding's authentication system (both application and the server) or to create an own ad hoc solution. The* EVALUATION ENGINE *can be run on the (well protected)* MAIN SERVER *of the subsidiary or on a separate* EVALUATION SERVER.

In the example we have a number of outsourcing relations. The subsidiary decides that it is more profitable to store private user data on an external database: EXTERNAL DATA DB1 *or* EXTERNAL DATA DB2. *The whole* EXTERNAL RATING *service (managing the activity, application and the node) is outsourced to one of two available* CREDIT BUREAUS. *In contrast to the outsouring the* EXTERNAL RATING, *outsourcing of* AUTHENTICATION *to the holding company (application and server) does not include the activity itself because it is the subsidiary which is responsible for performing all required operations (e.g., collect authorization data). Also the* PAYMENT *services are entirely provided by other subsidiaries of the holding which are spread around the city and easily available for customers.*

Note, that the Enterprise Model captures the hierarchy of a BP (even to the level of its atomic activities) but it does not provide information about the control flow of the process. Contributions of activities to the chosen BP depend on the control flow and we need a more explicit way (a Business Process model) to model these contributions.

3.1 Security requirements in the models

Enterprise and BP models are used for the elicitation and specification of security requirements for the modelled elements. The elicitation starts with the identification of business security objectives (BSOs) (or business security goals). The objectives are made specific for the chosen BP and become security requirements. Then for each level of hierarchy of the BP the requirements are specified for each activity at this level. By hierarchy, here and in the sequel, we mean hierarchy determined by structural activities of the BP ("sequence", "parallel", "loop", "choice"). Such a hierarchy can be easily obtained from a non-hierarchical representation of the BP.

In this paper we consider that there is only one requirement corresponding to one business objective. In other words, each element (activity, node or application) in the model has as many security requirements as many security objectives have been defined in the beginning (in contrast to [5]).

When the lowest hierarchy level of the BP has been reached (all atomic activities) we get the starting requirements in the Enterprise Model. The decomposition of requirements continues according to the Enterprise Model (similar to [5]): first requirements for information objects and components are identified whose violation could impact the previously identified requirements for the activities. Then, security requirements for nodes are specified. At the end, the threats which impact the requirements are identified. It is naturally to identify several threats for the same requirement for a model element, in contrast to identification of requirements where only one requirement for an element is identified for one security objective. These threats may be the cause of violations of any requirement for an element in the Enterprise Model.

1. Incorrect fulfilment of duties by employees and managers violates requirements on the business layer.

2. The threats impacting applications required for the activities violate requirements for components. Internal administration staff (also former administrators) are more probable potential attackers at this level. Skilled outsiders, which can find and exploit vulnerabilities by themselves, also should be considered.

3. The threats impacting physical security and system software required for correct operation of nodes (e.g., operating system, database, etc.) are identified on the physical layer. At this level outsiders are the main attackers, though also insiders should not be neglected.

Example 4 *The general requirement for the BP is the* prevention of fraud. *Going down through the hierarchy of requirements we determine the threats for activities (e.g.,* Rating manipulations *for* INTERNAL RATING *activity,* commit fraudulent transaction *for payment by* CASH*), data (e.g.,* unauthorized modification *of* USER'S DATA*), components (e.g.,* corrupt rating engine*), nodes (e.g.,* capture control *over* RATING SERVER*). The following threats are identified for nodes:* worms, viruses, hacker attacks *(attacks using various exploits),* Trojan horses. *Components can be affected by back doors,* phishing *and* password guessing *attacks on the components requiring authentication and* advance hacker attacks *with home-made application-specific exploits. The business elements can be affected by a dishonest employee* abusing his power *(e.g., not performing internal rating check or deliberately modifying data).*

Note, that though, in theory, some requirements can be connected with themselves (i.e., there are circles) these situations are not very practical and are not considered in this paper. On the business level the circle could be formed by ”input” and ”output” dependencies (see Figure 2) between Business Processes and Information Objects. Although a situation in which a BP compromises an information object and then, processing the information object compromises itself, is possible, we consider such a situation as improbable. A circle may be also formed by a link between several nodes (e.g., network connection) but data for our analysis (Section 5) is more likely to be taken from security logs of the nodes and thus we *already* have the total number of attacks for the node.

4 Protection Appraisal DAG

In order to estimate the assurance of a BP we need a suitable data structure derived from the Enterprise and BP models. The mathematical structure we use for our approach is Protection Appraisal DAG (Directed Acyclic Graph). In [16] we described the process in details and provided algorithms for building the Protection Appraisal DAG using a BP as well as for the reconstruction of the BP. In this work we extend the model on the architecture of the enterprise.

Initially, a Protection Appraisal DAG ($\mathcal{PAD}$) is built from a BP specified in the extended BPMN (as it is done in Figure 2). In short, for each activity we add a node (called appraisal node) which denotes the security requirement for this activity. The top node is a general node corresponding to the BP as a whole. The appraisal nodes which correspond to the activities of sub-processes (source set) are connected to the (target) appraisal node for the decomposed activity with a decomposition edge. If we have several alternatives (several sub-processes) to fulfill the same activity we draw several decomposition edges leading to the same target node starting from different source sets. In case an activity is outsourced we add an additional node and connect it with the appraisal node for the outsourced activity. This is done because the expected values of assurance indicators depend on the trust levels of the partners.

A similar strategy holds for converting the Enterprise Model. All enterprise elements are represented with appraisal nodes. Several appraisal nodes (source nodes) are connected with a target node with one

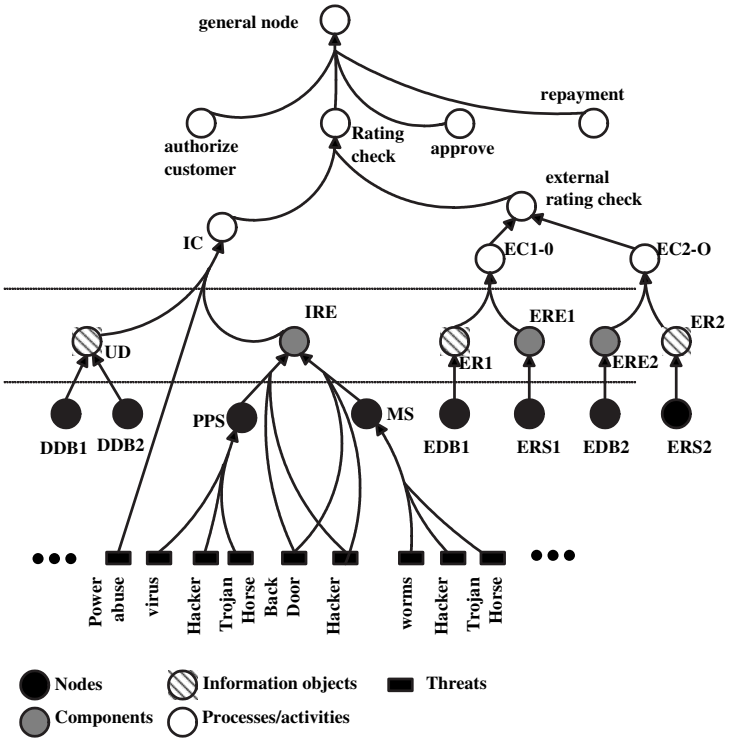

Figure 4: Protection Appraisal DAG.

decomposition edge if the elements corresponding to the source nodes are required for operation of the element corresponding to the target node.

Example 5 *In Figure 4 we show the part of the Protection Appraisal DAG built for the* RATING CHECK *sub-process. The nodes are colourized differently according to their type of enterprise model element to enhance clarity. To clarify the mapping between Figure 2 and Figure 4 we marked nodes by abbreviating their names (e.g., the nodes corresponding to* INTERNAL RATING ENGINE *are marked as* IRE*). All nodes denoting outsourcing relations have suffix "-O", e.g.,* EXTERNAL CHECK 1 *and* 2 *are denoted by EC1-O, EC2-O. We did not illustrate all threats but only those relevant for* INTERNAL RATING *activity,* INTERNAL RATING ENGINE *and two servers where the engine may run. Other threats should be attached similarly. Note, that the threats are connected with the target node directly for enterprise nodes only (e.g., for* MAIN SERVER *(MS)) while for other elements (impacted also by lower elements) the contributions of the threats are attached to the hyperedge leading to the target node (e.g., for* INTERNAL RATING ENGINE *(IRE)).*

In general each source node in the Protection Appraisal DAG contributes differently to the appraisal of a target node. The different contributions can be captured by assigning several weights to each decomposition edge[2]. For the sake of simplicity, we sometimes just say "assign weights" to an edge when we mean that several sets of weights should be assigned to the edge. Each weight has its own meaning depending on the dependency it belongs to.

[2]In fact, we have to use a hypergraph-like structure (called FD-graph) if we want to attach several weights to one edge, but this is not important for the purpose of this paper.

Relation	Formula	Meaning
$flow$, $sequence$	$w_i = t_{q_i}/t_q$	Relative time of execution. t_{q_i} - average time for execution of an activity i; t_q - average time of execution of the target activity (whole sub-process)
$choice$	$w_i = p_i * t_{q_i}/t_q$	Relative time of execution multiplied by the probability to choose branch i
$loop$	$w = 1$	No change of the indicator.
$outsourcing$	$w = 1/T_p$	T_p - trust level of partner p.

Table 1: Weights for structural activities

Definition 1 *A* Protection Appraisal DAG *($\mathcal{PAD}$) is a quadruple $\langle Q, E, F_e, L_e \rangle$ where Q is a set of appraisal nodes of a BP and an Enterprise Model and E is a set of decomposition edges. Each decomposition edge is an ordered pair $\langle S_q, q \rangle$ from an arbitrary nonempty set $S_q \subseteq Q$ (*source set*) to a single node $q \in Q$ (target node). L_e is a set of edge-dependant labelling functions which assign a set of vectors of weights to each edge. F_e is a set of edge-dependant propagation functions which compute the indicators for a target node taking as arguments indicators for source nodes.*

A classical problem in the graph theory is finding the "shortest", i.e. optimal, path. In our approach the "shortest" hyperpath determines the architecture with the highest assurance (with the best values of the assurance indicators). In our previous work [15] we proposed a polynomial algorithm for finding the shortest path using a monotone function as an aggregation function. This algorithm works with only one requirement/indicator. In particular we used a weighted function: The function for an edge $e = \langle S, q \rangle$:

$$F_e = \sum_{\forall q_i \in S} w_i * I_{q_i} \tag{1}$$

where I_{q_i} means the value of indicators for the node q_i belonging to the source set S; w_i is a corresponding weight for the edge. In this work we also use the same weighted function applied for each requirement separately. The weights used for the aggregation function for a BP have the following meanings presented in Table 1.

For all enterprise elements a weight may be seen as the multiplication of two probabilities. Thus the weight for an edge e for a node i and a requirement j will be:

$$w_j^i = p_{req,j} * p_{el}^i;$$

where $p_{req,j}$ - the probability to impact the considered requirement and p_{el}^i - the probability that the considered element will be impacted (but not another one). p_{el}^i may be seen in most cases as a relative time of execution of a higher layer element element using a lower layer one (e.g., the relative time of execution of INTERNAL CHECK on MAIN SERVER) which can be taken from business process descriptions.

Many of the components can be reused. p_{el}^i component is the same for all components of the same vector of weights. $p_{req,j}$ is domain independent and can be used for all vectors on the same level for the same requirement. p_{el}^i component for all threats is equal to 1.

5 Assessment

After creation of the Protection Appraisal DAG the analyst identifies the values of leaf appraisal nodes, i.e., the values of assurance indicators for all threats. The data are received from history records or estimated

by security experts. Note that, monitoring events on the business layer require trusted logging procedures in order to have reliable logs. If the activity is outsourced to a subcontractor the values are taken from the corresponding contract.

Now we have a classical problem of finding the "shortest" path: the root set Q' is a set of all leaf appraisal nodes (corresponding to all possible threats) and the target is the top node, which is the general node denoting the quality of protection for the whole process. Our algorithm presented below searches for a multi-objective shortest path (and values) with non-superior functions in a hypergraph using pareto-optimal principle. Unfortunately, the multi-objective optimal path problem is not polynomial, though the algorithm may be significantly optimized [9]. On the other hand, we claim that the algorithm proposed below is polynomial in depth and in number of paths because it is based on the our algorithm from [15].

The main problem with multi-objective cases is that we cannot make an unambiguous decision while comparing two vectors of values (value vector, in the sequel).

Example 6 *Imagine that we evaluate two security objectives:* prevent fraud *and* keep the loan process confidential. *We cannot make an optimal decision while choosing the best host for the* INTERNAL RATING ENGINE *if it is known that: (i) the* POST-PROCESSING SERVER *is more probable to be used for corrupting the* INTERNAL RATING ENGINE *than the* MAIN SERVER, *but (ii) it is less used for compromising confidentiality of the results of the rating check rather than the* MAIN SERVER.

The best thing we can do is to use the Pareto optimality principle to compare vectors. We should compare all elements in two vectors one by one. If *all* values in one value vector are bigger than in the other one we can make an unambiguous decision eliminating the former vector from further consideration. Note, that such decision can be made only if the aggregation function is monotone in all its variables (e.g., weighted function). If at least one element in the first vector is less than the corresponding element in the second one then we should propagate both vectors. These vectors are called non-dominated, because non of them dominates others.

Below we present the algorithm for computing all non-dominated paths. The algorithm is based on [15] and adapted for multi-objective optimal shortest path problem using [14].

In the algorithm by indicator we mean the triple: $i = \langle V, e, 2^I \rangle,\ i \in I$ where V is value vector, e - the last traversed edge, and 2^I is a set of indicators of source nodes of the edge e which were used for calculation of the value vector V. We also assume that all nodes are marked with numbers of edges leading to them and all edges are marked with number of source nodes.

The main Algorithm 1 works similar to the one presented in [15] but the calculation of indicators and their comparison should be specified for a multi-requirement analysis.

We will get several possible indicators by computing value vectors (Algorithm 2). The number of possible indicators we get is equal to the multiplication of the number of indicators for the source nodes. In order to store the path each indicator stores the last edge and all indicators used for calculation. The indicators must be checked for dominance and all dominated indicators must be removed (Algorithm 3).

The algorithm aggregates the contributions caused by identified threats and selects the branches which have the non-dominated security indicators. The results of the run of the algorithm is twofold: (i) we receive the set of non-dominated security vectors for the architecture as a whole; (ii) the set of optimal hyperpaths which indicate the most secure system architectures.

At the end of propagation we can compare all non-dominated vectors using ALE analysis [10] and choose the best one, and the best hyperpath correspondingly.

Algorithm 1 Optimal multi-objective hyperpath

Require: $\mathcal{PAD} = \langle Q, E, F_e, L_e \rangle$: Protection Appraisal DAG;
I_{Leaf} : set of indicators for leaf threats;
Ensure: I : real; set of indicators for all nodes
1: Assign maximum indicators to appraisal nodes;
2: Assign premise indicators (I_{Leaf}) to leaf appraisal nodes;
3: Add leaf appraisal node to a working set;
4: **while** working set is not empty **do**
5: Take randomly a node (q') from the working set;
6: **for** each outgoing edge from q' ($\langle S_q, q \rangle \ . \ q' \in S_q$) **do**
7: Mark the node as visited;
8: **if** All source nodes from S_q are visited **then**
9: Compute Indicators($\langle S_q, q \rangle$);
10: Mark the edge as traversed;
11: **if** all edges leading to q are traversed **then**
12: Choose Non-Dominated Indicators(q);
13: Add q to the working set;

Algorithm 2 Compute Indicators

Require: $\langle S_q, q \rangle$ edge
Ensure: $I(q)$ set of indicators for node q
{calculate all possible sets of indicators}
1: **for** all possible combinations of indicators of source nodes **do**
2: Compute the value vector for an indicator;
{store the hyperpath for the indicator}
3: Assign $\langle S_q, q \rangle$ and used indicators of source nodes to the indicator.

6 Related works

A business oriented perspective on security has long been argued for by many authors in the field [11, 22]. Combining model-based approaches with risk and security methods [13, 19] has in this regard been a path followed by a variety of researchers and practitioners in the last years.

An academic approach that is following a model-based risk analysis is CORAS [3]. CORAS approach uses models mainly for descriptive purposes and to visualize and communicate security aspects to various stakeholders. In contrast we use models to depict dependencies and enhance them with a mathematical model to assess the overall security of the architecture.

Suh and Han [23] use a business model to identify business functions in order to evaluate the relative importance of information assets for these functions. Suh and Han focus solely on the security requirement of operational continuity. For this purpose they use a measure to denote the relative importance of technical assets for business functions.

Morali et al. are using a model-based approach assess confidentiality [18]. Their approach is also based on a model of the architecture and the aggregation of values along the identified dependencies. Our work is different with regard to the weights assigned to these dependencies.

Jürjens [8] has developed the UMLSec approach for model-based security engineering based on UML. UMLSec is an extension of UML that can be used to express security relevant information in UML diagrams of a system. The approach is mainly targeted towards secure system development while we use a enterprise modeling to analyse functional dependencies between business and technical artifacts.

Algorithm 3 Choose Non-Dominated Indicators

Require: q - evaluated node
Ensure: $I(q)$ - Indicators for q
{remove all dominated indicators}
1: **for** all indicators of q **do**
2: **if** indicator is dominated **then**
3: Remove the indicator

From a security evaluation perspective our work is close to analysis of attack trees. In attack trees the main goal of an attacker (can be seen as violation of the general requirement) is hierarchically decomposed on more concrete steps, fulfilment of which could lead to accomplishing the general goal [21, 17]. Using Enterprise and BP models for construction of Protection Appraisal DAG make goal decomposition more objective. We also provide a quantitative analysis which is based on a well-known risk assessment technique.

In contrast to [2] we first aggregate values (number of attacks in a period of time, which is similar to Annual Rate of Occurrences) and then estimate the impact for business security objectives rather than consider threats separately without linking them with the objectives.

Clark et. al. [6] also aggregated risk levels using an attack tree in order to evaluate the impact on enterprise's mission (business goal), but for aggregation the authors used number of existing vulnerabilities on leaf nodes treating them as risk measures which differs from classical Risk Assessment. Also none of the attack tree based evaluation approaches helps to choose the more secure architectural design.

7 Conclusion

In this paper we presented the approach which outlines how using Enterprise and BP models can help to select the most secure architecture for a BP among several alternatives. The approach also takes into account that some parts of the process or architecture may be outsourced to external partners which may have different trust levels. The analysis considers evaluation using several requirements at once. We supported our approach with algorithms for the aggregation of indicators and selecting the non-dominated architectures. The analysis eliminates all dominated alternatives and returns only the architectures which cannot be easily compared. For such alternatives we proposed to use the well-known ALE analysis to make an unambiguous decision.

References

[1] ACFE. *The 2006 Report to the Nation.* Association of Certified Fraud Examiners, 2006. available via `http://www.acfe.com/documents/2006-rttn.pdf`.

[2] S. Bistarelli, F. Fioravanti, and P. Peretti. Defense trees for economic evaluation of security investments. In *Proc. ARES*, 2006. IEEE Computer Society.

[3] F. den Braber et al., Model-based security analysis in seven steps — a guided tour to the CORAS method. *BT Technology Journal*, 25(1):101–117, 2007.

[4] R. Breu and F. Innerhofer-Oberperfler. Model based business driven IT security analysis. In *Proc. SREIS*, August 2005.

[5] R. Breu, F. Innerhofer-Oberperfler, and A. Yautsiukhin. Quantitative assessment of enterprise security system. In *Proc. WPA*. 2008.

[6] K. Clark, J. Dawkins, and J. Hale. Security risk metrics: Fusing enterprise objectives and vulnerabilities. In *Proc. IAW*, 2005.

[7] IT Governance Institute *Control Objectives for Information and related Technology (COBIT)* Version 4.1. 2007. available via `www.isaca.org/cobit/`.

[8] Jürjens, J. Model-Based Security Engineering with UML *FOSAD 04/05*, Springer, 42-77, 2005.

[9] M. Ehrgott and X. Gandibleux. A survey and annotated bibliography of multiobjective combinatorial optimization. *OR Spectrum*, 22(4):425–460, 2000.

[10] L. A. Gordon and M. P. Loeb. *Managing Cybersecurity Resources: a Cost-Benefit Analysis*, 2006.

[11] G. Herrmann and G. Pernul. Viewing Business Process Security from Different Perspectives. In *Proc. of 11th Int.l Bled Electronic Commerce Conf.*, 1998.

[12] F. Innerhofer-Oberperfler and R. Breu. Using an enterprise architecture for IT risk management. In *ISSA*, 2006.

[13] S. A. Kokolakis, A. J. Demopoulos, and E. A. Kiountouzis. The use of business process modelling in information systems security analysis and design. *Information Management & Computer Security*, 8(3):107–116, 2000.

[14] E.Q.V. Martins and J.L.E. dos Santos. The labeling algorithm for the multiobjective shortest path problem. Technical Report 99/005, CISUC, 1999.

[15] F. Massacci and A. Yautsiukhin. An algorithm for the appraisal of assurance indicators for complex business processe. In *Proc. QoP.*, 2007.

[16] F. Massacci and A. Yautsiukhin. Modelling of quality of protection in outsourced business processes. In *Proc. IAS*, 2007.

[17] S. Mauw and M. Oostdijk. Foundations of attack trees. In *Proc. CISC*, 2005.

[18] A. Morali et. al., IT Confidentiality Risk Assessment for an Architecture-Based Approach. *Proc. BDIM*, 2008.

[19] M. zur Muehlen and M. Rosemann. Integrating Risks in Business Process Models. *Proc. ACIS*, 2005.

[20] OMG (Object Management Group). *Business Process Modeling Notation Specification*, 1.0, 2006. available online.

[21] B. Schneier. Attack trees: Modelling security threats. *Dr. Dobb's journal*, December 1999.

[22] B.v. Solms and R.v. Solms. From information security to...business security? *Computers & Security*, 24(4):271–273, June 2005.

[23] B. Suh and I. Han. The IS risk analysis based on a business model. *Information & Management*, 41(2):149–158, 2003.