A General Method for Assessment of Security in Complex Services *

Leanid Krautsevich[1], Fabio Martinelli[2], and Artsiom Yautsiukhin[2]

[1] Department of Computer Science, University of Pisa, Pisa, Italy
krautsev@di.unipi.it
[2] Istituto di Informatica e Telematica, Consiglio Nazionale delle Ricerche, Pisa, Italy
{fabio.martinelli,artsiom.yautsiukhin}@iit.cnr.it

Abstract. We focus on the assessment of the security of business processes. We assume that a business process is composed of abstract services, each of which has several concrete instantiations. Essential peculiarity of our method is that we express security metrics used for the evaluation of security properties as semirings. First, we consider primitive decomposition of the business process into a weighted graph which describes possible implementations of the business process. Second, we evaluate the security using semiring-based methods for graph analysis. Finally, we exploit semirings to describe the mapping between security metrics which is useful when different metrics are used for the evaluation of security properties of services.

Keywords: business processes, services, semirings, risk, security metrics, design graph.

1 Introduction

Rapidly changing world requires rapidly changing solutions. This is one of the reasons why service oriented technologies (Grid, Web Services, Cloud) become so popular. The idea behind such technologies is to be agile, easily reconfigurable and provide different alternatives to fulfil the same goal. Thus, service consumer is able to select the alternative she likes the most, i.e., the service which has the most suitable qualities, expressed as Service Level Agreement (SLA).

Security requirements also must be included in the agreement, in order to protect valuable assets not only during data transmission, but also during data usage [12, 13]. Naturally, selection of the most suitable business process must take into account security requirements. Usually, requirements, or policies (we use terms requirements and policies interchangeably) are precisely expressed with help of metrics, which indicate the quantity of some parameter. We assume that metrics may be evaluated using statistical methods, intrusion detection systems, using questionnaires, or simply assigned by security specialists [14, 15].

* This work was partly supported by EU-FP7-ICT NESSoS and EU-FP7-ICT ANIKETOS projects.

Service consumer is able to select the service which has the best metric values. The problem appears as soon as we have a complex service, *a business process*, which is composed of several simple services. A way to aggregate the values of simple services is required for the evaluation of the complex service. Moreover, existing alternatives of the implementation of a business process should be compared and the optimal alternative should be selected. Such analysis is useful not only for service consumers, but also for the service orchestrator which provides the complex service hiding the implementation details. For example, instead of selection of the most secure implementation, the orchestrator may find the level of protection it is able to guarantee even in case of problems with some simple services. Finally, the method for the analysis should be independent from the metric used for the evaluation, since simple services may be evaluated using different security metrics[3].

1.1 Contributions

The main contributions of this paper are the following.

- Provide a general method for aggregation of security metrics and selection the most secure implementation of a business process. This goal is achieved using a special mathematical structure "semirings".
- We have shown how similar metrics could be combined to conduct a general analysis. This goal is achieved by considering relations between metrics using mapping between semirings.

The paper is structured as follows. Section 2 presents a primitive transformation of a business process described in Business Process Modelling Notation into a graph. In Section 3, we evaluate overall security of a business process analysing the graph with help of semiring-based methods. Section 4 shows how the relation between security metrics may be described. Section 5 is devoted to the related work. Section 6 presents directions for future work and a conclusion.

2 Decomposition of a Business Process into a Design Graph

We consider a general business process (complex service) composed of simple abstract services. An abstract service describes a single job that should be done during the execution of the business process. Many notations for description of the business processes could be used as a starting point for the analysis. For example, Business Process Execution Language (BPEL) [1] is one of the most well-known and wide-spread notations. The main disadvantage of using BPEL for our purpose is that this language requires too much low-level details, which are not used for the analysis. On the other hand, the process can be

[3] We admit, that security is not the only quality which must be taken into account during selection of the best alternative, but in this article we focus only on security.

described with Business Process Modelling Notation (BPMN) [2]. BPMN is a high-level notation and, thus, is suitable for the analysis of high-level security properties. On the other hand, it is also only a graphical notation and an ad-hoc formalisation is required for automatic transformation of the model.

We follow BMPN for the description of a business processes. We consider a business process which is composed using the four basic structured activities: sequence, choice, flow, and loop. *Sequence* describes a situation when the services or structured activities are executed sequentially. *Choice* allows selecting a service on the basis of attributes of the business process or events external to the business process. *Flow* is used to denote two or more services or activities run in parallel. *Loop* supports the iterative execution of services and activities.

We extend this set with one more structured activity called *design choice* similarly to Massacci and Yautsiukhin [17, 16], which denotes a design alternative for a business process. Design alternatives denote sub-processes which fulfil the same functional goal, but in different ways (i.e., these are different sub-processes). The alternatives provide different qualities in general, and security properties in particular. Semantics of the design choice is similar to a regular choice, but the design choices are solved during the implementation of the business process, while the regular choice is solved during the execution. We exploit a gateway with letter "X" inside to denote the regular choice and a gateway with letter "D" inside to denote the design choice in a business process diagram.

Each abstract service has several real instantiations, *concrete services.* Concrete services are run by different service providers. For instance, an on-line trading platform may be provided by Amazon or eBay, off-the-shelf e-mail solution by Gmail or Hotmail. We suppose that an orchestrator of a business process signs a contract with each service providers that deliver concrete services for the implementation of the business process. The contract is based on a service level agreement proposed by a service provider and accepted by a service consumer. The orchestrator determines the security level of each concrete service analysing the policies. The security level is computed as a security metric which the orchestrator exploits for the future security evaluation and selection of the business process implementations. An essential goal of the orchestrator is to solve all design choices and select instantiations for the abstract services in a way to obtain the most secure implementation of the business process.

Example 1. We consider an on-line shop as an example of the business process (see Figure 1). First, a customer uses an on-line engine for searching and selecting items for buying. The owner of the shop would like to choose the way to implement the on-line engine. She considers two alternatives: to buy an on-line trading platform or to rent a server and install a content management system (CMS) there. Second, selected items are paid using a payment service. Third, items are shipped to the customer. Finally, the customer gets information about the payment and conditions of shipping by e-mail or VoIP service. The owner considers two opportunities to organise an e-mail service: to run an off-the-shelf e-mail solution or to organise her own e-mail server buying a hosting and installing an e-mail server software.

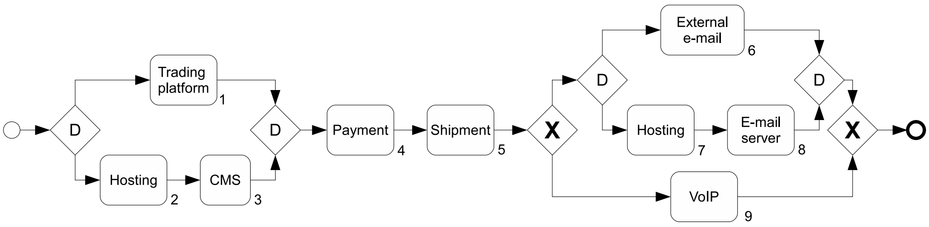

Fig. 1. Example of a business process in BPMN: an on-line shop

2.1 A Mathematical Model

We make a mathematical model of a business process in order to use it later for our analysis. A high level business process can be easily transformed into a graph in several ways (e.g., [17, 16]). We propose first to make the transformation into a process algebra. We use a notation similar to Calculus of Communicating Systems (CCS) [19]. Then we build the graph according to the execution flow.

In the process algebra there are several operators, which are useful for formalising a process. Let a_i be an abstract service, S^A be the set of all jobs (abstract services) in a business process such that $a_i \in S^A$. P and Q are two processes consisting of actions combined with basic operators and terminated with $\mathbf{0}$. *Sequence* activity can be formalised as $a_i.P$, i.e., first action a_i is executed and then process P. Parallel activity *Flow* is coded as $P|Q$. *Non-deterministic choice* is formalised as $P+Q$, i.e., process P or process Q is executed.

In the current work we simplify the transformation assuming that an orchestrator has information about usual execution of business process in advance. Thus, all *choices* except design choices are known in advance and we can consider only a part of the initial business process containing design choices only. *Loop* activity is considered as a number of the same executions in a raw. We assume that the orchestrator knows exact number of loops or uses the average number. The following technique is used to obtain a graph after the transformation of the business process into the process algebra.

We call a *Design Graph* a graph composed of concrete services connected with edges representing message flow in a business process. The root node of the graph is an empty node representing the beginning of a business process. For the sequential composition, the child of a node is the next executed service in a process algebra description. In case of parallel composition we select any activity first and then another one, hence, the parallel composition is a sequence of nodes in the graph. Intuition behind such transformation is that we consider the security of a business process and all parallel branches should be successfully executed for the successful execution of the business process. Regular choices are solved according to assumption above. A node has several outgoing edges if corresponding service is followed by a design choice. We call such node an "or-node". Outgoing edges lead to nodes corresponding to the first services in design alternatives grouped by the design choice. In addition, "or-node" is used to represent a choice between concrete services. Finally, an empty node is used to conclude the graph. The direction of connections is the same as the direction of message flow in the business process diagram. Moreover, each node is assigned with a weight according to the value of a metric expressing service security. Source node and final nodes have zero weights. Now, we are able to formalise the Design Graph we receive after transformation of a business process description.

Definition 1. *Let $S^A = \{a_i\}$ be a set of abstract services. Let also $S^C = \{c_{ij}\}$ be a set of concrete services and any $c_{ij} \in S^C$ is a j-th concrete service for an abstract service a_i. Then, we define Design Graph as a tuple $\langle N, E, L\rangle$. Where*

- *$N = \{n_{ij}\} \cup \{n_0\} \cup \{n_\infty\}$ is a set of nodes, where nodes n_{ij} correspond to the concrete services c_{ij}, n_0 and n_∞ are initial and final nodes corresponding to the start and the end of the business process;*
- *E is a set of edges between nodes which correspond to the message flow in the business process;*
- *$L : N \mapsto A$ is a labelling function which assigns to every node a number from the domain of a security metric, the source node and the final node are always assigned with zero value of the metric.*

Example 2. We continue Example 1. Consider transformation from a business process in Figure 1 into a Design Graph. Suppose the owner of the on-line shop knows that most of her customers prefer to be contacted via e-mail. This information helps an orchestrator of the business process to remove the exclusive choice between a VoIP service and implementation of e-mail service on the final step of the business process.

The design graph starts with the initial node n_0 which has three children n_{11}, n_{12}, and n_{21}. Nodes n_{11} and n_{12} describes the selection between concrete services instantiating a trading platform in Figure 1 (e.g., n_{11} is for Amazon and n_{12} is for eBay). The alternative implementation of the on-line engine is presented by node n_{21} which stands for a hosting service and two its children n_{31} and n_{32} denoting CMSs. Nodes n_{41} and n_{42} represent payment services, n_{51} stands for shipping service, n_{61} and n_{62} denote external mailing services, n_{71} and n_{72} represent hosting for an e-mail server, n_{81} and n_{82} are the e-mail server software. The graph ends with the node n_∞ which stands for the end of the business process. We display the resulted Design Graph produced from the business process in Figure 2.

Now if we follow the mathematical model every implementation of the business process is represented as a path in the Design Graph.

Definition 2. *Let $\langle N, E, L\rangle$ be a Design Graph and $n, n', n'' \in N \ \wedge \ N' \subseteq N \ \wedge \ E' \subseteq E$. A path from n to n' is a sub-graph $\pi_{\langle n,n'\rangle} = \langle N', E', L'\rangle$ such that*

1. *$N' = \{n\}$ and $E' = \emptyset$ if $n' = n$;*
2. *$N' = \{n'\} \cup N''$ and $E' = \{\langle n', n''\rangle\} \cup E''$, where $\langle N'', E'', L''\rangle$ is a path $\pi_{\langle n,n''\rangle}$;*

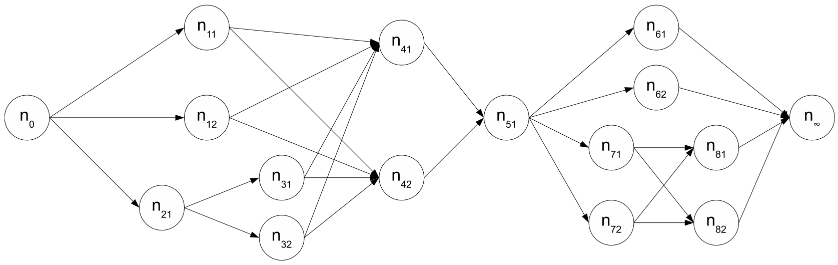

Fig. 2. A design graph representing an on-line shop

3. $L' \equiv L$.

Any path $\pi_{\langle n_0,n_\infty\rangle}$ *represents implementation of the business process, where* n_0 *is the initial node and* n_∞ *is the final node.*

In addition we define set $P(n_0, n_\infty) = \{\pi_{\langle n_0,n_\infty\rangle}\}$ representing all the possible paths between n_0 and n_∞. Each path has its own weight obtained by aggregating weights of nodes belonging to the path. The weight of the path is representing the security metric for an implementation of a business process. Aggregating of weights corresponds to aggregating of metric values. The problem of the selection of the most suitable implementation of the business process can be seen as *to find such path in a Design Graph that the weight of the whole path is the best one (e.g., maximal or minimal) among all possible.* We call the path with optimal value of metric the *shortest path* and denote it as $\pi^S_{\langle n_0,n_\infty\rangle} \in P(n_0, n_\infty)$. Implementation of the business process corresponding to the shortest path has the best value of the security metric. This implementation is the most secure one.

3 Security-aware Selection of a Business Process Implementation

As soon as the Design Graph is built we can start analysing it in order to select the implementation of a business process which satisfies the desirable customer's policies. First, we simplify the task and select the most secure business process implementation among other alternatives. Naturally, if this selection does not satisfies the desirable customer's policies then no other implementation does.

We aim at the assessment of the security of a business process using different security metrics. However, in this section, we assume that the security of all concrete services is assessed using the same security metric. This assumption will be relaxed in Section 4. Each node n_{ij} in a Design Graph is assigned with weight $w_{ij} = L(n_{ij})$. The initial n_0 and the final node n_∞ are assigned with a zero value. We look for a method that allows abstracting the security metrics and using universal algorithm for computation of the shortest path.

Mehryar Mohri [20] proposed a framework that contains algorithms for searching for the shortest path in a weighted graph, extending the work of E. Dijkstra

[9]. The framework exploits the notion of *semiring* for the abstraction of weights and operators over weights. A semiring consists of a set of values D (e.g., natural or real numbers), and two types of operators: aggregation ($\otimes$) and comparison ($\oplus$) values and constraints. Formally, the semiring is defined as follows [4]:

Definition 3. Semiring T *is a tuple* $\langle D, \oplus, \otimes, \mathbf{0}, \mathbf{1}\rangle$*:*

- D *is a set of elements and* $\mathbf{0}$, $\mathbf{1} \in D$*;*
- $\oplus$*, is an additive operator defined over (possibly infinite) set of elements* D*, for* $d_1, d_2, d_3 \in T$*, it is communicative* $(d_1 \oplus d_2 = d_2 \oplus d_1)$ *and associative* $(d_1 \oplus (d_2 \oplus d_3) = (d_1 \oplus d_2) \oplus d_3)$*, and* $\mathbf{0}$ *is a* unit *element of the additive operator* $(d_1 \oplus \mathbf{0} = d_1 = \mathbf{0} \oplus d_1)$*.*
- $\otimes$ *is a binary multiplicative operator, it is associative and commutative,* $\mathbf{1}$ *is its* unit *element* $(d_1 \otimes \mathbf{1} = d_1 = \mathbf{1} \otimes d_1)$*, and* $\mathbf{0}$ *is its* absorbing *element* $(d_1 \otimes \mathbf{0} = \mathbf{0} = \mathbf{0} \otimes d_1)$*;*
- $\otimes$ *is distributive over additive operator* $(d_1 \otimes (d_2 \oplus d_3) = (d_1 \otimes d_2) \oplus (d_1 \otimes d_3))$*;*
- $\leq_T$ *is a partial order over the set* D*, which enables comparing different elements of the semiring, the partial order is defined using the additive operator* $d_1 \leq_T d_2$ *(*d_2 *is better than* d_1*) iff* $d_1 \oplus d_2 = d_2$*.*

The weight $\delta^S(\pi^S_{\langle n_0, n_\infty\rangle})$ of the shortest path $\pi^S_{\langle n_0, n_\infty\rangle}$ is computed using additive operator $\oplus$:

$$\delta^S(\pi^S_{\langle n_0, n_\infty\rangle}) = \bigoplus_{\forall \pi_{\langle n_0, n_\infty\rangle} \in P(n_\infty, n_0)} \delta(\pi_{\langle n_0, n_\infty\rangle}) \qquad (1)$$

Here $P(n_0, n_\infty)$ is the set of all paths from the initial node n_0 to the final one n_∞. The cost $\delta(\pi_{\langle n_0, n_\infty\rangle})$ of the path $\pi_{\langle n_0, n_\infty\rangle}$ is computed using multiplicative operator:

$$\delta(\pi_{\langle n_0, n_\infty\rangle}) = \bigotimes_{\forall n_{ij} \in \pi_{\langle n_0, n_\infty\rangle}} w_{ij} \qquad (2)$$

We need to express security metrics as *semirings* for exploitation of universal algorithms for the search of shortest path in a weighted graph.

3.1 Semirings for Expressing Security Metrics

Security of the business process is assessed using security metrics. Different semirings must be used to express different metrics. In the following list we display several semirings and describe metrics expressed using these semirings.

- Weighted semiring $\langle R^+, min, +, \infty, \mathbf{0}\rangle$ represent the *risk of a successful attack* on a business process. A path in a tree computed under preferences using weighted semiring will minimize the overall sum of risks of successful attacks on services composing the business process. We assume that the business process is compromised if a successful attack compromises at least one service included in the business process.

- Probability semiring $\langle [0,1], max, \times, \mathbf{0}, \mathbf{1} \rangle$ expresses the *probability of a successful operation* of the business process (a resistance to all attack). In case we know the probability p_i of compromising the i^{th} service, then $(1 - p_i) \in [0,1]$ is the probability to tolerate all attacks.
- Semiring $\langle N^+, min, +, \infty, \mathbf{0} \rangle$ serves for identification of a path with the *minimal number of attacks.*

This is not a complete list of metrics and semirings that can be used for the searching a way for the optimal execution of a business process. Other semirings can be defined for other metrics if necessary. Note, that semirings serve also for the assessment of non-security properties of the business process. For instance, semiring $\langle N^+, min, +, \infty, \mathbf{0} \rangle$ is used for identification of the minimal number of steps to reach the goal of the business process. Semiring $\langle R^+, max, min, \mathbf{0}, \infty \rangle$ allows evaluating the *latency* of the business process if we assume that only one delay may occur during the business process execution. Probability semiring $\langle [0,1], max, \times, \mathbf{0}, \mathbf{1} \rangle$ may be used to express users *trust* to the business process.

We are able to apply any semiring-based algorithm (e.g., Generic Single Source Shortest Distance [20]) for searching of the shortest path after a semiring was chosen and the problem is defined by Equations 1 and 2. Note, that the algorithm uses the weights on the edges while we use the weights on the nodes. The algorithm can still be applied if we use the weights for the node as the weight of of every incoming edge leading to this node.

Example 3. Suppose each concrete service is assessed with the quantitative risk value. Weighted semiring $\langle R^+, min, +, \infty, \mathbf{0} \rangle$ is used to represent the risk. There are 48 possible paths in the graph presented in Figure 2. Without loss of generality, we consider just two paths for shorter explanation. Let weights of nodes be $w_{11} = 100$, $w_{41} = 120$, $w_{51} = 150$,$w_{61} = 90$, $w_{62} = 110$, $w_0 = w_\infty = 0$. First, we find the weights for paths $\pi^1_{\langle n_0, n_\infty \rangle} = n_0 n_{11} n_{41} n_{51} n_{61} n_\infty$ and $\pi^2_{\langle n_0, n_\infty \rangle} = n_0 n_{11} n_{41} n_{51} n_{62} n_\infty$. The weights $\delta^1(\pi^1_{\langle n_0, n_\infty \rangle}) = 480$ and $\delta^2(\pi^2_{\langle n_0, n_\infty \rangle}) = 500$ are computed using multiplicative operator $\oplus$ of weighted semiring. Then the best weight is selected using additive operator min: $\delta^S = min(\delta^1, \delta^2) = 480$. The shortest path is $\pi^S_{\langle n_0, n_\infty \rangle} = \pi^1_{\langle n_0, n_\infty \rangle}$. Note, that we used a simplified computation for this example, when the mentioned algorithms (e.g., [20]) are much more efficient.

The idea of exploitation of semirings has several advantages. The first advantage is that it allows re-evaluating of security of a business process and choose an alternative implementation of the business process. The need of the alternative implementation may be caused by the change of the security level of current implementation or by the change preferences of an orchestrator. The second advantage is that the orchestrator can evaluate the business process using different security criteria and select several implementations corresponding to different security metrics. The orchestrator can exploit an implementation satisfying the major part of criteria.

4 Interoperability of Services

Our idea requires services being assessed using the same metric. However in the real world a situation when security of all services is evaluated using the same metric is not always possible. Also a service consumer may express her security requirements using metric different than service provider's one. For instance, consider a situation when the security of the first part of services is assessed using minimal number of attacks and the security of the second part is assessed using risk. One more example is a situation when the service provider assesses risk level using quantitative risk while service customer uses qualitative risk scale. There is a need for a method that can evaluate the security in case of several metrics. We propose to tackle the issue by mapping between security metrics. The metrics used for an evaluations of services may be mapped to the most general one (e.g., risk) on the basis of formal relations between metrics considered in [14, 15]. The relations may be expressed using mappings between semirings presented by Bistarelli et al. in [3]. The analysis described in Sections 2 and 3 should be applied after the mapping is done.

The approach for the mapping between semirings is proposed for abstracting soft constraints in constraint satisfaction problems (CSPs). The idea behind the mapping is that two CSPs problems H and $\widehat{H}$ have mutual solutions if constraints are the same but expressed by different semirings.

According to [3] the mapping between semirings is done as follows. Suppose there are two semirings $T = \langle D, +, \times, 0, 1\rangle$ and $\widehat{T} = \langle \widehat{D}, \widehat{+}, \widehat{\times}, \widehat{0}, \widehat{1}\rangle$. Our goal is to map the first semiring onto the the second one. A Galois insertion $\langle \alpha, \gamma\rangle : \langle D, \leq_T\rangle \rightleftharpoons \langle \widehat{D}, \leq_{\widehat{T}}\rangle$ is used for the mapping. Here α and γ are two mappings such that α and γ are monotonic. If there is a problem H over semiring T we get a problem $\widehat{H} = \alpha(H)$ over semiring $\widehat{T}$ applying α. Mapping allows evaluating bounds for the solution of H if the solution of the problem $\alpha(H)$ is known. If there is the problem H over T, and $\widehat{h}$ is an optimal solution of problem $\alpha(H)$ with semiring value $\widehat{d}$ in $\alpha(H)$ and d in H, then there is an optimal solution h of H with semiring value $\overline{d}$ such that $d \leq \overline{d} \leq \gamma(\widehat{d})$.

A problem of searching a shortest path in a graph is a CSPs problem [20]. Thus, we are able to find bounds for a weight of the shortest path in a Design Graph if we do mapping between metrics using semirings. The bounds may be used as an approximated value of the security of business process. The bounds also may be used as a starting point for searching a precise value.

Example 4. In the example, we suppose that some services are evaluated using quantitative risk and others are evaluated using qualitative risk. The mapping between quantitative and qualitative risks is presented in Figure 3.

Qualitative and quantitative risks can be expressed as semirings. Weighted semiring $\langle R^{+}, min, +, \infty, \mathbf{0}\rangle$ serves for the assessment of quantitative risks. A semiring $\widehat{T} = \langle \widehat{D}, \widehat{+}, \widehat{\times}, \widehat{0}, \widehat{1}\rangle$ is used for qualitative risk assessment. Here $\widehat{D} = \{low, medium, high\}$, $\widehat{+}$ is an additive operation that chooses minimal risk value (min), $\widehat{\times}$ is a multiplicative operation choosing maximal risk value (max), $\widehat{0} =$

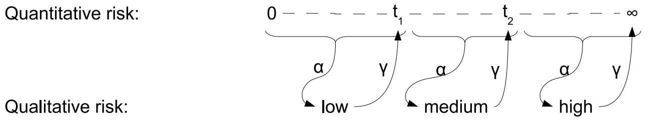

Fig. 3. An example of a mapping between quantitative and qualitative risks

high, $\widehat{1} = low$. We suppose that $high \leq_{\widehat{T}} medium \leq_{\widehat{T}} low$, which means that a *low* risk is better that *medium* and *high* risk, and a *medium* risk is better than a *high* risk. Other qualitative scale can be used. If there are M values in a qualitative scale, we need to divide interval R^+ into M smaller intervals, thus we need to determine $M - 1$ thresholds t_i. The map α translates the intervals into qualitative values. The map γ translates qualitative risk values into thresholds.

5 Related Work

The process of selection the optimal business process must also be based on the quality of protection as one of the essential criteria. Such claim has been recently raised by various authors [11, 12].

The first problem here is to find a method of security assessment suitable for services. Henning [10] proposed to evaluate a service against 15 security domains each of which is evaluated separately and a level (from 1 to 4) is assigned to it. Casola et al. [6] proposed a method for selection of the best alternative based on the distance between two lists of security levels using the assessment results provided by the method of Henning. In another work Casola et al. [5] proposed a more generic method for aggregation of different appraisals common for security and quality of service. Another approach to security assessment of a service is to use risk for aggregation. For example, Krautsevich et al. [13] proposed to assess risk of satisfaction of every security policy and then find the overall risk. The overall risk is then used to select the most secure service.

The second problem in assessment of services is to find the quality of protection for a complex service, i.e., service which consists of other services. Cheng et al. [7] proposed a framework for aggregation of downtime of a BP. The authors consider the BP as a set of services, rather then as a structured sequence of steps. Derwi et al. [8] analysed security in pervasive computing using multi-objective optimisation. The aim of the analysis was to analyse the workflow in order to select a set of security solutions. Our goal is slightly different, we focus on selection the best alternative. Moreover, we consider a more complex scenario, when nodes have some complex value, in contrast to the work of Derwi et al., where 0-1 metrics were considered.

The closest approach to our work is the work of Massacci and Yautsiukhin [17, 16]. The authors proposed an approach which transforms a business process to a tree and selects the most secure alternative according to the defined aggregation functions. In our work, we proposed a different way of graph construction and, more important, generalised the problem using semirings. Although, the method proposed by Massacci and Yautsiukhin is able to solve wider range of problems, our current proposal is based on a well-developed mathematical structure (semirings) and, thus, automatically allows applying different existing algorithms for analysis. Moreover, semirings allow considering interoperability of services assessed using different security metrics.

Similar problems have been considered in a non-security domains. For example, Jeager et al. [18] provided several aggregation functions for such criteria as minimal execution time, cost, etc. Yu et al. [21] proposed a method for selection of alternative business processes using the graph theory. These works do not consider security assessment. Moreover, our goal is to propose a generic framework which can be applied to different metrics (satisfying the required conditions).

6 Conclusion

This paper is the first attempt to assess the security of business processes using semirings. We described a simplified decomposition of a business process into a design graph. We considered computing security metric values for implementations of business process and selecting of the best implementation on the basis of semiring-based methods for weighted graphs analysis. We provided the idea for mapping between security metrics for the case when security properties are expressed using different metrics.

We would like to notice that this paper is just an initial step towards the assessment of the security using semirings. We are going extend the method in several ways. First, we are going to relax the assumptions on the transformation of a business process into the a design graph, since orchestrator often does not have an information to avoid choices and loops. Second, we will aim at expressing more metrics as semirings. Third, we will try to determine explicitly mappings between security metrics on the basis of their formal relation. Fourth, we are going to focus on the case when parameters of the business process are dynamic. In particular, we are going to consider the cases when the security level of concrete services may change or a monitoring system returns different values, rather than the ones declared in SLAs. Another example could be a new concrete services added to the business process on-the-fly or, vice versa, become unavailable. These cases require re-evaluation of the affected part of the process and may result in selection of another composition. Finally, we would like to implement the method as a software prototype and evaluate its performance.

References

1. Business process execution language for web services version 1.1, 2003. Available via http://public.dhe.ibm.com/software/dw/specs/ws-bpel/ws-bpel.pdf on 13/04/2011.
2. Business process model and notation (bpmn) version 2.0, January 2011. Available via http://www.omg.org/spec/BPMN/2.0 on 19/05/2011.

3. S. Bistarelli, P. Codognet, and F. Rossi. Abstracting soft constraints: Framework, properties, examples. *Artificial Intelligence*, 139:175–211, 2002.
4. S. Bistarelli, U. Montanari, and F. Rossi. Semiring-based constraint satisfaction and optimizatio. *J. ACM*, 44(2):201–236, March 1997.
5. V. Casola, A. R. Fasolino, N. Mazzocca, and P. Tramontana. An ahp-based framework for quality and security evaluation. In *Proceedings of 12th IEEE International Conference on Computational Science and Engineering*. IEEE, 2009.
6. V. Casola, A. Mazzeo, N. Mazzocca, and M. Rak. A SLA evaluation methodology in Service Oriented Architectures. In *Proceedings of the 1st Workshop on Quality of Protection.*, Milan, Italy, 2005. Springer-Verlag.
7. F. Cheng, D. Gamarnik, N. Jengte, W. Min, and B. Ramachandran. Modelling operational risks in business process. Technical Report RC23872, IBM, July 2005.
8. R. Dewri, I. Ray, I. Ray, and D. Whitley. Security provisioning in pervasive environments using multi-objective optimization. In *Proceedings of the 13th European Symposium on Research in Computer Security*. Springer-Verlag, 2008.
9. E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1):269–271, 1959.
10. R. Henning. Security service level agreements: quantifiable security for the enterprise? In *Proceedings of 1999 workshop on new security paradigms*. ACM, 2000.
11. C. Irvine and T. Levin. Quality of security service. In *Proceedings of the 2000 Workshop on New security paradigms*. ACM, 2000.
12. Y. Karabulut, F. Kerschbaum, P. Robinson, F. Massacci, and A. Yautsiukhin. Security and trust in it business outsourcing: a manifesto. In *Electronic Notes in Theoretical Computer Science*, volume 179, pages 47–58, 2006.
13. L. Krautsevich, A. Lazouski, F. Martinelli, and A. Yautsiukhin. Risk-based usage control for service oriented architecture. In *Proceedings of the 18th Euromicro Conference on Parallel, Distributed and Network-Based Processing*. IEEE, 2010.
14. L. Krautsevich, F. Martinelli, and A. Yautsiukhin. Formal approach to security metrics.: what does "more secure" mean for you? In *Proceedings of the Fourth European Conference on Software Architecture: Companion Volume*. ACM, 2010.
15. L. Krautsevich, F. Martinelli, and A. Yautsiukhin. Formal analysis of security metrics and risk. In *Proceedings of Fifth Workshop in Information Security Theory and Practice*. Springer, 2011.
16. F. Massacci and A. Yautsiukhin. An algorithm for the appraisal of assurance indicators for complex business processe. In *Proceedings of the 3rd Workshop on Quality of Protection*. ACM, 2007.
17. F. Massacci and A. Yautsiukhin. Modelling of quality of protection in outsourced business processes. In *Proceedings of the The Third International Symposium on Information Assurance and Security*. IEEE, 2007.
18. G. R.-G. M.C. Jaeger and G. Mühl. QoS aggregation in web service compositions. In *Proceedings of the IEEE International Conference on e-Technology, e-Commerce and e-Service (EEE'05)*, 2005.
19. R. Milner. *Communicating and Mobile Systems: the pi-Calculus*. Cambridge University Press, 1999.
20. M. Mohri. Semiring frameworks and algorithms for shortest-distance problems. *J. Autom. Lang. Comb.*, 7(3):321–350, January 2002.
21. T. Yu and K.-J. Lin. A broker-based framework for qos-aware web service composition. In *Proceedings of the IEEE International Conference on e-Technology, e-Commerce and e-Service*. IEEE, 2005.