# Modelling Quality of Protection in Outsourced Business Processes *

Fabio Massacci
University of Trento
Fabio.Massacci@unitn.it

Artsiom Yautsiukhin
University of Trento
evtiukhi@dit.unitn.it

## Abstract

*There is a large number of research papers and standards dedicated to security for outsourced data. Yet, most papers propose new controls to access and protect the data rather than to assess the level of assurance of the whole process that is currently deployed.*

*The main contributions of the paper is an approach for aggregating security properties of individual tasks of a complex business process in order to receive the level of assurance provided by the whole process. The approach takes into account the fact that some tasks of a business process may be outsourced and thus account for not very reliable partners. The approach chooses the concrete business process offering the highest assurance among several possible design alternatives by building an optimal hyperpath traversing the business process.*

## 1 Introduction

The recent years have seen three major technological trends:

- service-oriented architectures and business process management platforms emerged as the architectures and technologies of choice.
- companies and institutions are often outsourcing the non-core part of business processes [10].
- the number and complexity of security and accountability requirements placed on business processes have increased (e.g. SOX, Basel II, etc.).

These trends have a profound impact on the trust models, security policies, procedures, and infrastructure that companies need to maintain [18, 17].

Research on workflow security ([2, 4, 16]) has traditionally focussed on access to data: it guarantees proper access to a single data along the workflow by enforcing dynamically a number of constraints on the users who can access that data. A typical constraint is Separation of Duty [23, 4, 2] so that a single malicious employee cannot corrupt the value of the data along the flow (e.g. by misappropriating company assets).

In the new context the problem is not whether a handful of sensitive data has been disclosed or separation of duty has been violated twice. In real systems, security breaches will always be present. What makes the difference is whether a whole process is *regularly* broken.

For negotiating the level of assurance of an outsourced process and demonstrating its compliance with security regulations we must appraise such "regularity". Indeed, in-house operations are often conducted in a best-effort manner, while outsourced services require a precise and detailed definition of the expected service level in terms of suitable security requirements. This level then will demonstrate the quality of protection of business process.

Unfortunately a business process may be very complex with re-outsourcing of some parts of the process. While it is often feasible to appraise the security of atomic activities (as the requirements are negotiated with the provider of the task or available from internal data), what is needed is the aggregation of this value in order to assess the level of assurance of the overall business process.

The main contributions of the paper is an approach for aggregating security properties of individual tasks of a complex business process in order to receive the level of assurance provided by the whole process. The approach takes into account the fact that some tasks of a business process may be outsourced and thus account for not very reliable partners. The approach chooses the most secure concrete business process among several possible design alternatives by building an optimal hyperpath for the business process.

## 2 Running example

Consider a bank holding company which outsources the concrete loan processing to several semi-independent subsidiaries. There is a specific subsidiary belonging to the

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

holding which provides loans. The subsidiary defined the business process shown in the Figure 1 to fulfil the assigned task. The process is depicted using BPMN (Business Process Management Notation) [20] which is a widely used notation. A simpler version of the scenario without outsourcing also described in [7, 23].

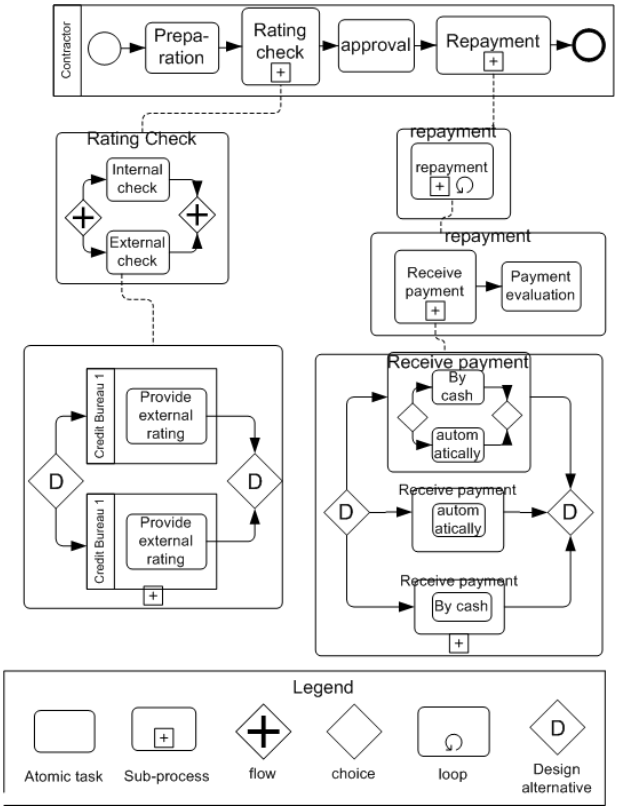


**Figure 1. Loan origination business process. Hierarchical view.**

The holding company is aware of a huge number of losses caused by frauds[1]. In the banking sector that we just described the typical problem is asset misappropriation: using organization's assets (especially cash) by an employee either directly or indirectly for the employee's benefit. Asset misappropriation accounted for approximately 90% of all frauds [1]. Therefore the holding wants to be sure that it is well protected against this type of losses.

## 3 Business Process and Outsourcing

At first, we identify the stake-holder entities in the outsourcing scenario.

[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]

**Definition 1** *A* Client *is an entity that interacts with a completed, self-contained business process. A* Contractor *is an entity which manages the business process and agrees to satisfy the client's requirements for such execution.*

Sometimes we use word *subcontractor* when an entity receives from another contractor a task assignment which is a part of a higher-level business process.

**Example 1** *In our running example the holding company plays role of client. To avoid the confusion with the holding company in the article we use term "customer" for the subject which wants to receive a loan. The contractor in the scenario is the subsidiary because it takes a responsibility to provide some service negotiated with the client (the holding company). Credit bureaus are subcontractors which provide a specific service (external rating check).*

Wlog, we assume that a contractor has a business process (BP) written in a hierarchical way using an extended BPMN notation. This means that a high level business activities decomposed by structural activities (i.e. sequence, choice, loop and flow) onto sub-processes. We connect collapsed and expanded sub-processes with thin dashed lines. So, if an activity is outsourced to a subcontractor it can be seen as further decomposed by the corresponding subcontractor. In a number of BP modelers (e.g., Maestro [11]) it is possible to represent process run by subcontractors by linking their "views".

Several design or deployment alternatives to fulfil an activity may exist which accomplish the same functional goals but provide different properties. We add a special construct to the notation to model design alternatives. At the end of modelling only one of the design alternatives should be left.

**Example 2** *There are three alternatives for the receive payment activity: only by cash, by withdrawing money from customer's account every month automatically or by giving both possibilities and allowing the customer to choose which option she prefers. It is known that payment by cash leads to more asset misappropriation cases than automatical withdraw when the whole transaction is accomplished by the bank's computer system alone.*

An important issue is to choose the subcontractors to which some parts will be outsourced. They provide different levels of assurance and have different levels of trust. These levels are not usually represented in BP model but only informally stated outside the model.

**Example 3** *Credit bureau CB1 claims that they have fewer asset misappropriation cases than CB2. On the other hand, it is known that in several cases CB1 failed to meet its claims. Thus the subsidiary trusts CB2 more than CB1 to meet the assurance level they claim to provide.*

To negotiate the assurance level of an outsourcing workflow it is necessary to set up some indicators (loosely speaking "metrics"):

**Definition 2** *Security indicators, define the* Quality of Security Services (QoSS)*, describe security qualities used by a contractor to achieve a high level of security. Assurance indicators, describe* Quality of Protection (QoP) *and are negotiated with the client to show that her security requirements are addressed.*

**Example 4** *The subsidiary measures its QoSS by percentage of compliance with a standard [15], number of people working on one transaction, frequency of audit, number of fraud patterns known by the automatic log analyzer.*

**Example 5** *The same subsidiary consider as a QoP (to be negotiated with the holding) the number of asset misappropriation per month. Other examples are mean time to intrusion affecting the holding customer's data [19], time spent after an undesirable event to restore the availability of the holding's data.*

Intuitively, QoSS describes functional requirements: what a contractor must *at-least* do. On the other hand, the natural intuition behind a QoP is that it describes negative events and specifies things that the contractor must *at-most* allow.

The contractor must map QoP to a functional QoSS to receive concrete requirements defining which security controls the contractor should install and which security policies to enforce. The mapping may be based on industry statistical data trends and company experience. We will not deal further with this issue here due to lack of space.

QoSS requirements for a whole BP cannot be aggregated because they specify how *separate activities* (or a group of activities) must be protected, and this normally happens by a variety of means [6]. On the other hand, external metrics specify protection of *client's data* which is processed by many activities and, therefore, can be aggregated. Therefore, we will consider QoP levels to establish the assurance offered by the overall system.

## 4 Protection Appraisal Dag

In order to compute the overall assurance indicators of a BP we introduce the formal notion of hypergraphs for appraisal node and will call it as *Protection Appraisal Dag*.

**Definition 3** *A* Protection Appraisal Dag *($\mathcal{PAD}$) is a triple $\langle Q, E, F_e\rangle$ where $Q$ is a set of nodes appraising a BP and $E$ is a set of decomposition edges. Each decomposition edge is an ordered pair $\langle S, q\rangle$ from an arbitrary nonempty set $S \subseteq Q$ (*source set*) to a single node $q \in Q$ (target node). $F_e$ is a set of edge-dependant propagation functions which compute the value of a target node taking as arguments values of source nodes.*

In the sequel, we will denote an appraisal node as $q$ and a decomposition edge as $e$. Capitals denote sets of the nodes and the decomposition edges. We also use $S$ for a source set of appraisal nodes.

One of the most important features of a $\mathcal{PAD}$ is a hyperpath which we define as follows:

**Definition 4** *Let $\mathcal{PAD} = \langle Q, E, F_e\rangle$ be a Protection Appraisal Dag, $Q' \subseteq Q$ be a non-empty subset of appraisal nodes, and $q$ be an appraisal node in $Q$. There is a* hyperpath *$h_{\langle Q',q\rangle}$ from $Q'$ to $q$ in $\mathcal{PAD}$ if*

1. *either $q \in Q'$*
2. *or there is a decomposition edge $\langle S, q\rangle \in E$ and hyperpath from $Q'$ to each appraisal node $q_i \in S$.*

The main goal is to find the minimal hyperpath, i.e., with the minimal value, from a starting set of nodes (leaf nodes) to some other node. There are a number of well known propagation functions for which the "shortest" path can be found in polynomial time (e.g. [3, 9]).

The typical example of propagation function is traversal cost [3]. Weights assigned to every edge denote the cost of traversing the edges. Each time an edge is traversed (i.e., added to the hyperpath) its weight is added to the overall value of the hyperpath. All values of the starting set nodes are assigned to zero.

**Definition 5** *The* traversal cost *$cost(h_{\langle Q',q\rangle})$ of an hyperpath $h_{\langle Q',q\rangle}$ from Q' to q is inductively defined as follows:*

1. *if $h_{\langle Q',q\rangle}$ is empty (i.e. $q \in Q'$) then:*

$$cost(h_{\langle Q',q\rangle}) \quad = \quad 0$$

2. *if the hyperpath $h_{\langle Q',q\rangle}$ has root $\langle S, q\rangle$ with subtrees $h_{\langle Q',q\rangle}, h_{\langle Q',q_1\rangle}, h_{\langle Q',q_2\rangle}, \ldots, h_{\langle Q',q_k\rangle}$, then:*

$$cost(h_{\langle Q',q\rangle}) \quad = \quad f_{\langle S,q\rangle}(cost(h_{\langle Q',q_i\rangle})|q_i \in S)$$

*In this setting the unique propagation function associated to the edge $e = \langle S, q\rangle$ and the set of appraisal node values $V_{q_i}$ ($V_{q_i} = cost(h_{\langle Q',q_i\rangle})$) relative to the source nodes $q_i \in S$ is*

$$f_e\left(\{V_{q_i}|q_i \in S\}\right) = w_e + \sum_{q_i \in S} V_{q_i} \qquad (1)$$

**Algorithm 1** From BP to Protection Appraisal Dag

**Require:** Business processes $BP$
**Ensure:** $\mathcal{PAD} = \langle Q, E \rangle$
1: Start with an empty set of edges (E)
2: Set of nodes (Q) consists of one top node
3: Set of activities ($Activity$) contains one top fictitious activity
4: **while** all activities from $Activity$ are not visited **do**
5: Extract an activity $A$ from $Activity$
6: **if** $A$ is delegated **then**
7: **for** each delegation **do**
8: add a node to $Q$ denoting outsourcing
9: add an edge from this node to the node for A($q_A$) to $E$
10: add $A$ (without outsourcing) to $Activity$
11: **end for**
12: **else** {if $A$ is decomposed by the same partner}
13: **for** each alternative **do**
14: add nodes for activities of the sub-process $BP_{sub}$
15: add an edge from these nodes to the node $q_A$ to $E$
16: add all activities of the sub-process to $Activity$
17: **end for**
18: **end if**
19: **end while**
20: Generate Backward mapping table;

## 5 Back and forth BP and QoP

Initially, a $\mathcal{PAD}$ is built from a BP specified in the extended BPMN. Algorithm 1 shows the process for building $\mathcal{PAD}$ . At the end we use a function to generate backward mapping table which stores the information required for the reconstruction of the BP once an assurance optimal configuration (design choices and partners selection) has been done with $\mathcal{PAD}$ .

Figure 2 illustrates the result of such construction for our running example.

**Example 6** *The sequence of high level activities is shown as a decomposition of a top appraisal node into four appraisal node s. Then the appraisal node s for the activities are further decomposed. The first and the third activities are atomic and are not decomposed. For each credit bureau we have a separate decomposition edge: only one of them should be selected and will contribute to the target appraisal nodeThe same holds for design alternatives for* receive payment*: only one strategy for repayment will be chosen.*

The propagation functions depend on used assurance indicators and on type of decomposition edges (four for structural activities and one for outsourcing relations).

**Example 7** *For such assurance indicator as "number of assets misappropriation cases per month" we can define the following propagation functions for a decomposition edge*

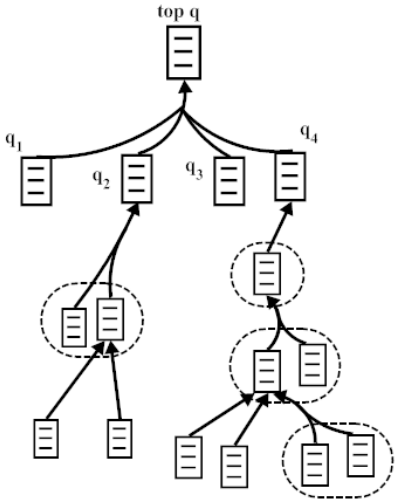


**Figure 2. Protection Appraisal Dag.**

$e = \langle S, q \rangle$ *corresponding to a "flow" node:*

$$f_{e_{flow}} = \sum_{\forall q_i \in S} w_i * V_{q_i} \tag{2}$$

*The constant in the formula is a weight* $w_i = t_{q_i}/t(q)$ *where* $t_{q_i}$ *is average time for execution of an activity* $q_i$ *and* $t(q)$ *is mean time for execution of the target activity (i.e., all source activities). Note, that the sum of the weights for all source activities is* not *equal to 1 because the activities are fulfilled simultaneously and the sum of execution times is greater then the time of execution of the target activity.*

**Example 8** *The function for a decomposition edge* $\langle \{q\}, q \rangle$ *corresponding to an outsourcing edge* $e$ *for the same assurance indicator can be the following:*

$$f_{e_{outsourcing}} = w * V_q \tag{3}$$

*The constant in the formula is a weight* $w = 1/T_p$ *where* $T_p \in [0; 1]$ *is a level of trust of partner p. If the value is 1 the contractor trusts the partner to meet the agreed security properties completely. If the value is 0 the contractor does not trust the partner at all, i.e., he believes that any agreed protection will not be met.*

After some processing we should reconstruct the BP which corresponds to a received $\mathcal{PAD}$ . For this purpose we need the information stored at the end of the Algorithm 1. Algorithm 2 shows the procedure for the reconstruction of the BP from a $\mathcal{PAD}$ .

**Algorithm 2** From Protection Appraisal DagTo BP

**Require:** $\mathcal{PAD} = \langle Q, E \rangle$, Backward mapping table
**Ensure:** Business processes $BP$

```
Start with empty BP;
Add the top appraisal nodein a working set (Rec) ;
while Rec is not empty do
    Pick one appraisal node q from Rec;
    Find an activity A associated with q;
    Find a subcontractor P to which A is delegated;
    if there is such subcontractor then
        Assign P to A;
        Add appraisal node (without outsourcing) to Rec;
    else {A is fulfiled by the same partner}
        Receive sub-process BP_sub for A
        if A is not an atomic activity then
            Add sub-process BP_sub for A to BP
            Add nodes associated with activities of BP_sub to Rec;
        end if
    end if
end while
```

# 6 Finding a Configuration with High Assurance

The contractor maps security functional properties of his controls to the values of leaf appraisal nodes. The mapping may be based on industry statistical data trends and personal experience. If an activity is outsourced to a subcontractor the values are taken from the contract. Finally, values for all leaf appraisal nodes (all atomic tasks) are defined.

Now we have a classical problem of finding the "shortest" hyperpath in a hypergraph for which efficient, polynomial time, algorithms have been proposed (e.g., [3]). Note, that these algorithms can be used only for those security parameters for which propagation functions are superior/inferior (e.g. number of attacks per execution). The traversal cost function (Equation 1) is a good example of a function for which shortest path algorithms work in polynomial time and that is usable in a number of practical cases.

Unfortunately, there are also some propagation functions that are appropriate for evaluating QoP assurance indicators which do not satisfy the conditions stipulated by hypergraph algorithms in the literature [3, 9].

**Example 9** *Suppose that for each activity in our scenario it has been found that the QoP assurance indicator expressed as "number of asset misappropriation cases per month" for* rating check *is 10/month (the maximum) and for* repayment *is 1/month (the minimum). The aggregated number of asset misappropriation cases is only 2/month because* rating check *is active during 5% of the observation period while* repayment *activity occupies about 90% of the observation period.*

This propagation function is neither superior (because the aggregated value is lower than the maximum value among the activities) nor inferior (it is also greater than the minimum value among the activities). Hence, we need to adapt the traditional hypergraph algorithms in order to find the optimal assurance solution in polynomial time.

The adaptation can be easily done for the security functions that are monotone with respect to each of its arguments (e.g., the one shown in Example 7). In this case the smaller value of some node in a hyperpath leads to the smaller value of the whole hyperpath. This allows us to unambiguously choose the best (smallest) alternative for the nodes where a decision should be made (node with several incoming decomposition edges) if values of all alternatives are known.

# 7 Related work and Conclusions

There is a large number of work done on access control in workflows. Botha and Eloff [5] provided a methodology for construction a "typed" role hierarchy for a workflow. Bertino at al. [4] formally expressed constraints on role assignment to tasks in a workflow and provided a planner which automatically assigns roles and users according to the constrains. Kang at al. [16] proposed a fine-grained and context-based access control mechanisms for inter-organizational workflows.

One of the first papers discussing security SLA in a large enterprise is [14]. The main idea is to check compliance the system with fifteen security domains split into best practices. For each best practice the security service level is determined and added to the SLA (yet it does not consider task outsourcing). Casola et al. [8] extended the security division to compare two SLAs or to find a security SLA which is the closest to the desired one. A similar idea of divide-and-conquer technique was applied to evaluation of Web Service security in [25]. Gutierrez et al. [13] proposed a process for elicitation of security requirements, specification of security architecture and identification of security services to be implemented integrated in WS-based system development. Rodriguez et al. [21] proposed to determine which of three security services (authentication, encryption and access control) should be applied to each step of the BP and in such a way to elicit security requirements. Karjoth et. al. [18] claimed that security requirements must be reflected in the contract and their fulfilment must be somehow monitored.

In this work we have described the approach which helps a contractor to determine the most secure concrete BP among several design alternatives. We have given algorithms for converting a designing BP to Protection Appraisal Dag and back. This approach also captures level of trust between the partners and adjusts metrics accordingly. The designer must obviously identify manually the propa-

gation functions because they are related to the particular business instance. Once the functions are determined the reasoning algorithm will test all process configurations and determine the most secure one.

On the first glance the methodology seems to be very complex. On the other hand, the $\mathcal{PAD}$ may be constructed in polynomial time by a tool using the presented algorithms. Of course, each metric requires specification of the five propagation functions (four functions for BP structural activities and the one for outsourcing) but these function can be determined by security experts once and then used by security engineers. The premise values of leaf nodes are the input data for the approach and can be taken from statistics or from agreements between partners. The only set of parameters which security engineers should define by themselves are weights. This set is BP-dependent and is derived from specification of the process.

So far we have left out the issue of how the client can make sure that the argued QoP is actually enforced. The solution is using *Trusted Virtual Domains (TVDs)* [12] which are intended to connect a number of remote trustable virtual processing environments in one secure network. An alternative is presented in [22] where a trusted hardware component is embedded into the execution environment to verify the compliance of the system with an operational policy (which can be considered as a PLA). Skene et al. [24] provided a formally verifiable way of specifying an SLA which can be monitored.

The most important work to be done is the identification of the five propagation functions. In Examples 7 and 8 we have shown our specification of the functions for "number of attacks per period of time" security indicator. We are going to test the functions using statistics from the e-banking scenario in scope of SERENITY project. Briefly, the experiments will be conducted as follows. Knowing the QoPs of aggregated web services we will collect the statistics for each node in the $\mathcal{PAD}$ and compare (qualitatively evaluate the deviation) it with computed values.

## 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] V. Atluri. Security for workflow systems. *Information Security Technical Report*, 6(2):59–68, 2001.

[3] G. Ausiello, G. F. Italiano, and U. Nanni. Optimal traversal of directed hypergraphs. Technical Report TR-92-073, Pisa University and Monreal University, Berkeley, CA, 1992.

[4] E. Bertino, E. Ferrari, and V. Atluri. The specification and enforcement of authorization constraints in workflow management systems. *TISSEC*, 2(1):65–104, February 1999.

[5] R. A. Botha and J. H. P. Eloff. Designing role hierarchies for access control in workflow systems. In *Proc. of the 25th COMPSAC*, pages 117–122, 2001. IEEE Computer Society.

[6] S. A. Butler. Security attribute evaluation method: a cost-benefit approach. In *Proc. of the 22rd ICSE*, pages 232–240. ACM Press, 2002.

[7] S. Campadello et. al. A7.D1.1 Scenario selection and definition. Technical report, SERENITY project, 2006.

[8] V. Casola et. al. A SLA evaluation methodology in Service Oriented Architectures. In *Proc. of the 1st QoP.*, Milan, Italy, 2005. Springer-Verlag.

[9] G. Gallo et. al. Directed hypergraphs and applications. *Discrete Applied Mathematics*, 42(2-3):177–201, 1993.

[10] G. Goth. The ins and outs of it outsourcing. *IT Professional*, 1(1):11 – 14, 1999.

[11] U. Greiner et. al. A multi-level modeling framework for designing and implementing cross-organizational business processes. In *Proc. of the 1st TCoB*, pages 13–23. INSTICC Press, 2006.

[12] J. L. Griffin et. al. Trusted virtual domains: Toward secure distributed services. In *Proc. of the 1st HotDep*, June 2005.

[13] C. Gutiérrez, E. Fernández-Medina, and M. Piattini. PWSSec: Process for Web Services Security. In *Proc. of the 4th ICWS*, pages 213–222, 2006.

[14] R. Henning. Security service level agreements: quantifiable security for the enterprise? In *Proc. of the 1999 NSPW*, pages 54–60. ACM Press, 2000.

[15] E. Johansson and P. Johnson. Assessment of enterprise information security - an architecture theory diagram definition. In *Proc. of the 3nd CSER*, March 2005.

[16] M. H. Kang, J. S. Park, and J. N. Froscher. Access control mechanisms for inter-organizational workflow. In *Proc. of the 6th SACMAT*, pages 66–74, 2001. ACM Press.

[17] Y. Karabulut et. al. Security and trust in it business outsourcing: a manifesto. In *Proceedings of the 2nd STM. To appear.*, 2006.

[18] G. Karjoth et. al. Service-oriented assurance comprehensive security by explicit assurances. In *Proc. of the 1st QoP.*, September 2005. Springer-Verlag.

[19] B. B. Madan et. al. A method for modeling and quantifying the security attributes of intrusion tolerant systems. *Performance evaluatin journal*, 1-4(56):167–186, 2004.

[20] Object management group. *Business Process Modeling Notation Specification*, 1.0 edition, February 2006. available via `http://www.bpmn.org`.

[21] A. Rodrigues, E. Fernandez-Medina, and M. Piattini. Towards an integration of security requirements into business process modeling. In *Proc. of the 3nd WOSIS*, pages 287–297, 2005. INSTICC PRESS.

[22] A.-R. Sadeghi and C. Stüble. Property-based attestation for computing platforms: caring about properties, not mechanisms. In *Proc. of the 2004 NSPW*, pages 67–77, New York, NY, USA, 2005. ACM Press.

[23] A. Schaad, V. Lotz, and K. Sohr. A model-checking approach to analysing organisational controls in a loan origination process. In *Proc. of the 11th SACMAT*, pages 139–149, 2006. ACM Press.

[24] J. Skene et. al. The monitorability of service-level agreements for application-service provision. In *Proc. of the 8rd WOSP*, pages 3–14, 2007. ACM Press.

[25] Y. Wang and P. K. Ray. Evaluation methodology for the security of e-finance systems. In *Proc. of the 2005 EEE*. IEEE Press, 2005.