An Algorithm for the Appraisal of Assurance Indicators for Complex Business Processes*

Fabio Massacci Artsiom Yautsiukhin†
Dip. Informatica e TLC - Università degli Studi di Trento, Trento - Italy
Fabio.Massacci@dit.unitn.it, evthiuki@dit.unitn.it

ABSTRACT

In order to provide certified security services we must provide indicators that can measure the level of assurance that a complex business process can offer. Unfortunately the formulation of security indicators is not amenable to efficient algorithms able to evaluate the level of assurance of complex process from its components.

In this paper we show an algorithm based on FD-Graphs (a variant of directed hypergraphs) that can be used to compute in polynomial time (i) the overall assurance indicator of a complex business process from its components for arbitrary monotone composition functions, (ii) the subpart of the business process that is responsible for such assurance indicator (i.e. the best security alternative).

Categories and Subject Descriptors

K.6.5 [**Management of Computing and Information Systems**]: Security and Protection; C.4 [**Performance of Systems**]: *Measurement techniques; Reliability, availability, and serviceability*; G.2.2 [**Discrete Mathematics**]: Graph Theory—*Hypergraphs*

General Terms

Algorithms, Management, Measurement, Security.

Keywords

Assurance Indicator, Business Process, Hypergraphs, Quality of Protection, Security Indicator, Security Metrics.

1. INTRODUCTION

Some emerging trends are shaping the business of the future:

from a technological perspective highly dynamic service-oriented architectures (SOA) with a distributed security administration have emerged as the architectures of choice.

from a business perspective companies and institutions are outsourcing non-core parts of their business and their IT infrastructure. Outsourcing is iterated through subcontractors.

from a standards and regulatory perspective the complexity of requirements has increased especially with respect to security, privacy, and accountability.

Due to these emerging trends, companies can neither deliver nor accept best-effort software solutions that cannot be subject to independent audits. This applies to all software components and services, but in particular to the security solutions implementing the controls that are essential for auditing. In a nutshell *companies must provide certified assurance services to their customers and expect assured services from their contractors*. The American Institute of Public Accountants and the International Federation of Accountants started to address problems and opportunities in this area by developing best practices for "Assurance and Trust Services" for Web and IT applications[1] (see also [9]).

In order to provide certified assurance services we must first provide indicators that can measure security state of a process or the level of assurance available to another process. In other words we must realize in the security and trust domain those notions that are universally accepted in the management and accounting community to provide certifiable assurance of financial services and sound business risk management. Looking at CoBIT, the Information Systems Audit and Control Association's framework of indicators, processes and best practices for IT governance and control in companies [13], we find Key Goal Indicators (KGI) establishing measurable business objectives that must be reached to obtain successful services, whereas Key Performance Indicators (KPI) set up the measures on the business or technological infrastructure (such as the number of "negative" events) that allow evaluating the level of goal achievement. To account for the hierarchical structure of business processes high-level KGI/KPI can be mapped to KGI/KPI at lower business levels.

The notions that we need are the following ones [16]:

Assurance Indicator is a measurable indicator negotiated by a client and a contractor to show that the client's business assurance goals are addressed e.g. the number of attacks or breaches that affect the clients' assets.

Security Indicator measures technical security features used by contractors to achieve a high level of security, e.g. presence and quality of protection and regulatory models.

CoBIT's indicators and these indicators are dual in the same way that service engineering and security engineering are dual methodologies. Intuitively, a goal indicator points to a better business,

†Contact author.

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

QoP'07, October 29, 2007, Alexandria, Virginia, USA.

[1]See for example http://www.webtrust.org/overview.htm

thus a system with more features. Assurance indicators point to a more secure system, thus a system with less troubles. Performance indicators define events that are bad for business while a security indicator points to events that are good for security such as passing from RBAC to RBAC with separation of duty.

Unfortunately, having an indicator is not enough: business processes are complex and, to scale up to industry level case studies, we must be able to derive the global indicators for a process by combining the indicator for a global business process from its components. Further we would like to analyze several business process alternatives and choose the one which provides the best protection.

1.1 The contribution of this paper

In our previous paper [20] we introduced the notion of Protection Appraisal DAG and provided an algorithm for the construction of such DAG from a business process. Our initial assumption was that one could then use with little or no modification the polynomial time algorithms used for hypergraph. In particular, an algorithm for finding the "shortest" (optimal) path should aggregate indicators of atomic activities of a business process and select the more secure concrete business process among various design alternatives.

Unfortunately some natural practical assurance indicators require using non-superior/non-inferior functions (see Example 6 in Section 3) which are not amenable to existing efficient algorithms.

In this paper we show an algorithm based on FD-Graphs (a variant of directed hyper-graphs) that can be used to compute in polynomial time (i) the overall assurance indicator of a complex business process from its components for arbitrary monotone composition functions, (ii) the subpart of the business process that is responsible for such assurance indicator (i.e. the best security alternative).

2. OUTSOURCING LOAN PROCESSING

To make the discussion more concrete we start here with a running example[2]. Our case scenario is a bank holding company which outsources loan processing to semi-independent subsidiaries.

We start by defining a business process and its stakeholders:

DEFINITION 1. *Business Process (BP) is an ordered set of activities designed to produce required outputs. There are four types of such ordering (sequence, choice, flow (parallel execution) and loop) called structured activities.*

DEFINITION 2. *A* Client *is an entity interacting with a completed, self-contained BP. A* Contractor *is an entity managing the BP and agrees to satisfy client's requirements for such execution. A* subcontractor *is an entity that receives a subtask assignment, part of a higher-level BP, from another contractor.*

Here, the subsidiary executes the BP shown in Figure 1 to fulfil the assigned task. The BP is depicted using BPMN (Business Process Management Notation) [22], a widely used notation.

EXAMPLE 1. *The holding company is the client. The contractor in the scenario is the subsidiary because it takes a responsibility to provide some service negotiated with the client. Credit bureaus are subcontractors which provide a specific service (*external rating check*). To avoid the confusion with the holding company, we use term "customer" for the subject which want to receive a loan.*

Together with provision of a good quality of service (e.g. high response time) contractors should provide a good quality of protection for client's data (e.g. low number of viruses corrupting client's data). Obviously to provide such quality of protection contractors should implement a number of security controls and policies (e.g. install antivirus, enforce a policy for using e-mails). To assess the internal security of the BP the contractor can use what we called *security indicators* such as the frequency of anti-virus updates, the presence of sophisticated access control models and so on [16].

The key point is what indicators should be used by the client to assess that the appropriate level of quality of protection is in place. As argued in [16] and also in [12, Chap.3], internal security indicators are not appropriate for the client. It should rather use what we termed *assurance indicators*.

EXAMPLE 2. *The holding company is aware of a huge number of losses caused by asset misappropriation in this market[3] and therefore wants to be sure that the subsidiary is well-protected against these losses. So the assurance indicator can be the number of asset misappropriations which auditors have traced back to security failures in each BP activity.*

In our setting, in order to get an assurance indicator of the overall process we also need to account for the possibility of choice at design and deployment time. To this extent we added a construct to BPMN for modelling explicitly the design or deployment alternatives among BP activities which accomplish the same functional goals but have different qualities. At the end of modelling phase only one of the alternatives should be left.

EXAMPLE 3. *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.*

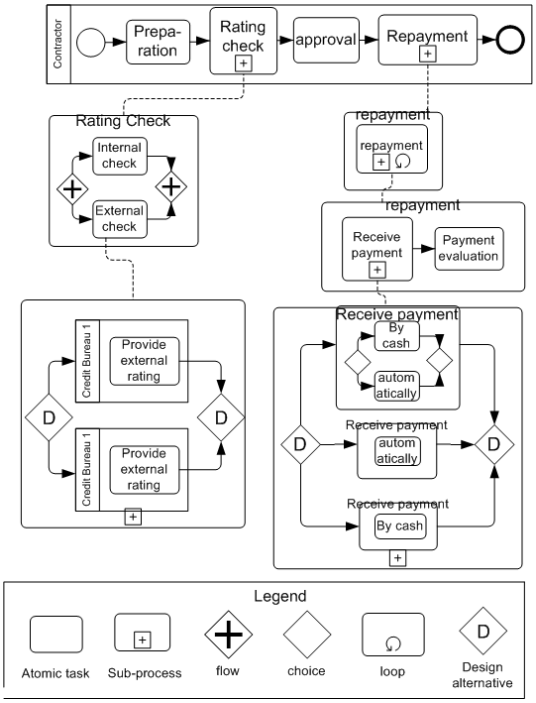

Figure 1: Example of a business process

[2]Additional details on a loan processing scenario are available on the SERENITY's site `www.serenity-project.org`.

[3]Asset misappropriation accounted for approximately 90% of all frauds, on average organizations loose 5% of annual revenue to asset misappropriation and median losses for banking companies are 258 000$ [1]

Another issue is the choice of the subcontractors to which some parts will be outsourced. These subcontractors provide different level of assurance and have different levels of trust. These levels are not usually represented in business process model but only informally stated outside the model. We also used dashed lines to show how sub-processes are expanded.

EXAMPLE 4. *An examples of deployment alternatives are two credit bureaus (CB1 and CB2) which provide the same service, i.e. trustworthiness rating of a client. Since it is well known that there were several cases when CB1 failed to meet its claims the subsidiary trusts CB2 more than CB1.*

3. Appraisal FD-Graph

In order to estimate the assurance level of a BP we need a data structure derived from the BP description in BPMN. Initially, a Protection Apraisal Dag ($\mathcal{PAD}$) is built from a business process specified in the extended BPMN. In [20] we described the process in details and provided algorithms for building the Protection Apraisal Dag as well as for reconstructing the "optimal" business process. In short, for each activity we add an appraisal node denoting the security requirement for the activity. The appraisal nodes corresponding to sub-processes (source set) are connected to the (target) appraisal node for the decomposed activity with a decomposition edge. If several alternative sub-processes can 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. In this way we can use the weight on the edge to account for the trust level of the subcontractor.

However, in practice we found out that each activity contributes differently to the appraisal of a target node. This requirement is not supported by the standard hypergraph notation which assign a single weight to a hyperedge. Therefore, we use a hypergraph-like structure called *Appraisal FD-Graph*. The Appraisal FD-Graph is also more convenient for the algorithm which we need for the quantitative analysis. We define Appraisal FD-Graph as follows.

DEFINITION 3. *Given domains $\mathbb{D}$ and $\mathbb{D}_2$ and given a Protection Apraisal Dag $\mathcal{PAD} = \langle Q, E, F_e \rangle$, where Q is a set of appraisal nodes and E is a set of decomposition edges. F_e is a set of edge-dependant propogation functions which compute values of a target node taking as arguments values of the source nodes. The* Appraisal FD-Graph *of $\mathcal{PAD}$ is a labelled graph $FD(\mathcal{PAD}) = \langle Q \cup C, E_q \cup E_c, F_c, L \rangle$, where:*

1. *$Q \equiv Q$ is a set of* appraisal nodes*;*
2. *C is the set of* compound nodes *which is in bijective relationship with E. If $\langle S, c_S \rangle \in E$ is a decomposition edge then c_S will denote the corresponding compound node, and any appraisal node $q_i \in S$ will be called a* component node *of the compound node c_S;*
3. *$E_q \subseteq C \times Q = \{\langle c_S, q \rangle | \langle S, q \rangle \in E\}$* full edge *, in bijective relationship with E.*
4. *$E_c \subseteq Q \times C = \{\langle q_i, c_S \rangle | c_S \in C$ and $q_i \in S\}$ compound edges connecting any compound node to its components.*
5. *$L : E_c \mapsto \mathbb{D}$ is a set of labelling functions which assign weights to compound edges.*
6. *$F_c : C \circ 2^{\mathbb{D}} \circ 2^{\mathbb{D}_2} \mapsto \mathbb{D}_2$ is a set of propogation functions which compute values of compound nodes taking sets of weights of compound edges and values of source nodes as arguments.*

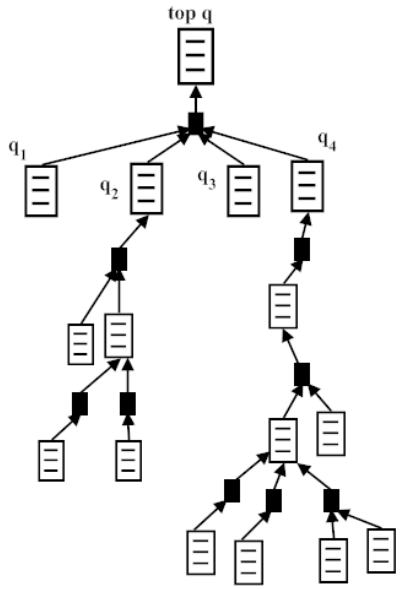

Figure 2: Appraisal FD-Graph

The choice of concrete function from the set depends on the type of the edge which is in a 1-1 relationship with compound nodes. In other words the functions F_e used in the Appraisal Dag map 1-1 with the functions used for the compound nodes F_c. The Appraisal FD-Graph for the running example is shown in Figure 2.

EXAMPLE 5. *The propagation functions correspond to the four BP structural activities ("sequence", "choice", "flow", "loop") and to the outsourcing relation.*

In Appraisal FD-Graph the notion of "path" is defined as follows:

DEFINITION 4. *Let $\mathcal{PAD} = \langle Q, E, F_e \rangle$ be a Protection Apraisal Dag, and $FD(\mathcal{PAD}) = \langle Q \cup C, E_q \cup E_c, F_c, L \rangle$ be the corresponding Appraisal FD-Graph. There is a FD-path $h^{FD}_{Q',x}$ from a source $Q' \subseteq Q$ to any target node $x \in Q \cup C$ if:*

1. *either $x \in Q'$*
2. *or $x \in Q \Rightarrow \exists\, x' \,.\, \langle x', x \rangle \in E_Q \ \wedge \ \exists\, h^{FD}_{Q',x'}$*
3. *or $x \in C \Rightarrow \forall\, x_i \,.\, \langle x_i, x \rangle \in E_C \ \wedge \ \exists\, h^{FD}_{Q',x_i}$*

A classical problem in the graph theory is finding the "shortest", i.e. optimal, path. In our approach the "shortest" FD-path determines the business process with the highest assurance (with the best value of the assurance indicator). For many propagation functions efficient algorithms already exists (e.g. for traversal cost [4]). Unfortunately, some practical and natural propagation functions for assurance indicators do not satisfy the conditions required by the hyper-graph algorithms in the literature [4]. These conditions are based on definition of *superior/inferior* function.

DEFINITION 5. *A function $g(x_1, ..., x_n)$ is a superior function if it is monotone nondecreasing in each variable and if $g(x_1, ..., x_n) \geqslant max((x_1, ..., x_n))$*

The notion of inferior function is obtained by the obvious swapping of less with greater and min with max.

EXAMPLE 6. *For each activity in our scenario the 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). However, 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 90% of the observation period.*

Since the result is less than the maximal value and greater than the minimal value of source nodes the natural propagation functions for the assurance indicator does not satisfy the condition. The corresponding propagation function is indicated below:

EXAMPLE 7. *The function for a node corresponding to a "flow" decomposition edge $e = \langle S, q \rangle$ could be*

$$F_{cflow} = \sum_{\forall q_i \in S} w_i * V_{q_i} \quad (1)$$

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

EXAMPLE 8. *The function for a node corresponding to an outsourcing edge $e = \langle \{q\}, q \rangle$ could be*

$$F_{cout} = w * V_q \quad (2)$$

The constant in the formula is a weight $w = 1/T_p$ where T_p is a level of trust of partner p. T_p belongs to $(0; 1]$ interval. If the value is 1 the contractor trusts the partner to meet the agreed security properties completely, when the value is 0 the contractor does not trust the partner at all, i.e., any agreed appraisal will likely be failed.

Hence, we need to adapt the traditional algorithms to find the optimal assurance solution in polynomial time.

4. ALGORITHM FOR MINIMAL FD-path

After creation of the Appraisal FD-Graph the contractor identifies the values of leaf appraisal nodes. In other words, it determines the values of assurance indicators for all atomic activities. The data are received from statistics available from external auditors or estimated by security experts if the activity is fulfilled by the contractor itself. If an activity is outsourced to a subcontractor the values are taken from the contract.

Now we have a classical problem of finding the "shortest" path: the root set Q_{Leaf} is a set of all leaf appraisal nodes and the target is the top node, which is the fictions node denoting the quality of protection for the whole process. As it has been shown in Section 3 for some propagation functions which are appropriate for assurance indicators the existing polynomial algorithms (e.g., [4]) are not applicable. Therefore, we need to create an algorithm which allows taking into account contribution of each activity and works with wider range of functions.

W.l.o.g. we consider one protection requirement (and one assurance indicator respectively) for each activity. The goal is to find the optimal executable business process leading to the minimal value of the requirement which can be met. In other words, we want to find a minimal FD-path in Protection Apraisal Dag from a set of leaf nodes to the top appraisal node and its value if values of assurance indicators for the leaf nodes are known. The algorithm can also be adopted to find the maximal FD-path, but it is doubtful whether the resulting value has some security interpretation [16].

Algorithm 1 Minimal FD-path

Require: $FD(\mathcal{PAD}) = \langle Q \cup C, E_q \cup E_c, F_c, L \rangle$: Appraisal FD-Graph;
V_{Leaf} : values of leaf nodes;
Ensure: $V[]$: real; {Values of assurance indicators}
$PATH[]$: decomposition edge; {FD-path}
1: Assign maximum value to simple appraisal node nodes
2: Assign premise values (V_{Leaf}) to leaf appraisal nodes;
3: Add leaf appraisal node nodes to a working set
4: **while** working set is not empty **do**
5: Take randomly a node (x') from the working set
6: **for** each outgoing edge from x' **do**
7: **if** reached node (x) is compound **then**
8: Mark the edge as traversed
9: **if** All source nodes leading to x are traversed **then**
10: Calculate value of node x ($V[x]$);
11: Add x to the working set
12: **else** {if it is a simple node}
13: Mark the alternative as reached
14: **if** all alternatives for x are reached **then**
15: Choose the minimal alternative
16: Add x to the working set
17: Store the alternative as path

Our algorithm extends the works of Ausiello et al. [4] and Gallo et al. [10] because the only requirement we impose on the propagation functions is that the functions is positive monotone in order to select the alternatives. Algorithm 1 informally describes the proposed procedure.

In order to make the things precise we use vector $SOURCE[]$ where number of incoming compound edges for each compound node is stored. This value is equal to the number of the source nodes of the decomposition edge with which the compound node is in bijection relation. Another auxiliary vector is $ALTERN[]$. It contains the number of alternative paths for each node appraisal node. This value is equal to the number of compound nodes which separately contribute to the appraisal node. $HEAP$ is used as a working set. Algorithms 2 shows the formal version of the algorithm.

We can prove the following properties of the algorihtms:

LEMMA 1. *It is possible to prove the following invariants:*

1. *Each node is visited*[4] *at most once.*
2. *Each edge is traversed at most once.*
3. *A node can be visited if and only if all nodes from which there is an edge leading to this node are visited.*
4. *After any number of execution of the "while" loop the set of traversed edges coincides with the set of outgoing edges for all visited nodes.*

To prove 1 we use induction for the proof. The leaf nodes cannot be visited more then once since there are no edges leading to them (only outgoing ones). For the inductive case we exploit the inductive hypothesis and rule out the possibility of vising a node twice (say before all its ancestors are visited and after all its ancestors are visited) by exploiting the conditions at line 11 (for compound nodes) and line 16 (for simple nodes).

For (2) an edge can be traversed only if its source node is visited. Moreover, it can be traversed only *once* when the node is visited.

[4]Visited node is a node at least once extracted from the HEAP

Algorithm 2 Minimal FD-path

Require: $FD(\mathcal{PAD}) = \langle Q \cup C, E_q \cup E_c, F_c, L\rangle$: Appraisal FD-Graph;
V_{Leaf} : values of leaf nodes;
Ensure: $V[]$: real; {Values of assurance indicators}
$PATH[]$: decomposition edge; {FD-path}
1: $V[Q] := \infty$; {simple nodes only}
2: $V[Q_{Leaf}] := V_{Leaf}$;
3: HEAP-insert(Q_{Leaf})
4: $ALTERN[Q] := |Incoming(Q)|$; {for all full nodes}
5: $SOURCE[C] := |Incoming(C)|$; {for all compound nodes}
6: **while** HEAP-nonempty **do**
7: HEAP-extract(x'); {randomly}
8: **for** $\langle x', x\rangle \in Outgoing(x')$ **do**
9: **if** {compound node} $x \in C$ **then**
10: decrement(SOURCE[x]);
11: **if** $SOURCE[x] = 0$ **then**
12: $V[x] = F_c(x; L(\langle x_1, x\rangle), ..., L(\langle x_q, x\rangle); V[x_1], ..., V[x_q])$; {$\langle x_i, x\rangle \in Incoming(x)$}
13: HEAP-add(x);
14: **else**
15: decrement(ALTERN[x]);
16: **if** $ALTERN[x] = 0$ **then**
17: **for** $x_i.\langle x_i, x\rangle \in Incoming(x)$ **do**
18: **if** $V[x] > V[x_i]$ **then**
19: $V[x] := V[x_i]$;
20: $PATH[x] = \langle x_i, x\rangle$;
21: HEAP-add(x);

Since each node can be visited only once (invariant 1) we deduce that also every edge is traversed at most once.

To prove 3, since each edge can be visited only once (invariant 2) and therefore counted only once then the conditions at lines 11 (for compound nodes) or line 16 (for simple nodes) hold only if *all* incoming edges are traversed. Since each edge is traversed only when its node is visited this means that all nodes from which there is an edge leading to considered node are visited.

Finally (4): when a node is visited its outgoing edges are traversed, so all traversed edges are outgoing from visited nodes.

Using these invariants we can prove the first result of this paper: since each node is extracted/added from/to the HEAP only once and the main loop (lines 6-21) terminates in time proportional to the number of edges and nodes.

THEOREM 1. *The algorithm terminates in polynomial time.*

A more precise calculation can be carried out by counting the individual contribution to the execution time.

The "for" loop (lines 8-21) terminates when all edges of a node are scanned. Inside the loop for each edge there is a comparison at line 9. Also for each $e \in E_c$ and $e \in E_q$ there are comparisons at lines 11 and 16 respectively. This builds up to $O(|E_c| + |E_q|)$.

For each appraisal node the minimal alternative is chosen (lines 17-20). The complexity is at most $O(|Q| * |C|)$. For each compound node a function F_c is computed only once (line 12). The complexity is $O(|C|) * O(|F_c|)$.

So, the practical complexity of the algorithm depends on the complexity of the propagation functions and is $O(|Q| + |C| + |Q| * |C|) + O(|C|) * O(|F_c|)$. For the propagation functions like Equations 1 and 2, where each source node is multiplied by the corresponding weight and then the results are summed up, the complexity of the function is maximum $O(|C| * |Q|)$ and the overall complexity is $O(|Q| + |C| + |Q| * |C|)$.

To show that the algorithm is also correct we first need a lemma:

LEMMA 2. *At the end of the execution of algorithm $Minimal\ FD - path$, any (simple and compound) node x has been visited if and only if there exists a FD-path from Q_{Leaf} to x in FD.*

To prove it, suppose that exists a FD-path $h_{Q_{Leaf},x}$ in FD but x is not visited by the algorithm. According to the recursive definition of FD-path 4, there exists at least one decomposition edge $\langle x', x\rangle \in h^{FD}_{Q_{Leaf},x}$ such that the node is visited by the algorithm while x is not. According to invariant 1 x has been added to the HEAP. This means that it has been visited, otherwise the algorithm should not terminate (Theorem 1).

For the only if case, before the execution of the "while" loop the visited nodes are Q_{Leaf} that means that there is an empty Protection Apraisal Dag from Q_{Leaf} to any of the nodes. When we extract node x there is a set of nodes $X = \{x_1, x_2, ...x_q\}$ among all visited nodes such that exist all required edges (all edges for a compound node and at least one edge for a appraisal node) from them to x. By inductive hypothesis there is a FD-path from Q_{Leaf} to every node x_i. By definition of a FD-path (Definition 4) $h_{Q_{Leaf},x} = h_{Q_{Leaf},x_1} \cup h_{Q_{Leaf},x_2} \cup ... \cup h_{Q_{Leaf},x_q} \cup \langle X, x\rangle$ is also a FD-path.

We can now state the second result of this paper:

THEOREM 2. *Algorithm Minimal FD-path computes correctly optimal paths from Q_{Leaf} to any node in Appraisal FD-Graph.*

Once again we use induction to prove that any visited appraisal node has *optimal* value. For the base case if a node is in Q_{Leaf}, its value is the value of the node itself and it is minimal.

Suppose now we are going to add a new node x' to the HEAP $X \subset Q$. By inductive hypothesis, any node $x_i \in X$ has an optimal value. Alternatives for node x' are computed using only values from this set ($x_i \in X \subset Q$). From this set we choose the optimal path $h_{X_{Leaf},x'}$. There are no other alternatives since in this case counter ALTERN for the node would not have been zero and the node would not have been added (invariant 3). So the optimal path for $h_{Q_{Leaf},x} = h_{Q_{Leaf},x_1} \cup h_{Q_{Leaf},x_2} \cup, ..., \cup h_{Q_{Leaf},x_n} \cup \langle X', x\rangle$ ($\{x_1, x_2, ..., x_n\} = X' \subset X$). Since the propagation function is positive monotone by the assumption the optimal value for the node is the minimal one.

5. RELATED WORK

The identification of a suitable indicator for assessment of security of a complex system is a well known problem. Some approaches suggested assigning a maturity level to the systems (e.g., the SSE-CMM), others recommended checking compliance with a security standard (e.g. ISO 17799) (e.g. [14]), third, calculated "mean-time-to-breach" indicator using vulnerability/attack graphs [19, 23]. The most popular approach nowadays is applying risk analysis for security evaluation [6, 25] which is based on economical assessment. In our work we propose an approach which aggregates indicators of simple elements rather than assessing the whole system at once.

There is a large number of articles about access control in workflows. Bertino at al. [5] formally expressed constraints on role assignment to tasks in a workflow in order to automatically assign roles and users according to the constrains. Kang at al. [15] proposed an fine-grained and context-based access control mechanisms for inter-organizational workflows. These papers do not discuss the issue of the quality of protection that a secure workflow may achieve and only have a 0-1 notion of security.

There are few works which deal with negotiation of security indicators between clients and contractors. One of the first works claiming that security requirements must be reflected in the contract is [17]. Casola [7] et. al. building upon [11] proposed to assess security of services separately within fifteen security domain and showed an algorithm to compare security SLAs against a target

SLA. A similar idea of divide-and-conquer technique was applied to evaluation of Web Service security in [26].

The closest work to our approach is [27]. The authors proposed to choose the concrete BP among several alternatives using provided qualities as the major criterium. However they restricted themselves to sequential decompositions of BP and the functions used for aggregation of qualities are suitable for existing hypergraph algorithms. Also the authors do not consider security/assurance indicators. Jeager et. al. [21] also provided several aggregation functions for a number of service qualities (e.g. minimal execution time, cost).

A directed hypergraphs introduced in [2] is a generalization of directed graphs which allows representing many-to-one relations. Classical problem of "finding a shortest path in a hypergraph" was studied by Ausiello [4] and Gallo [10]. The algorithms proposed by the authors are quite similar and based on the algorithm of Dijkstra [8]. Unfortunately the assumptions for the used functions in the hypergraphs are to strict to be applied for our work.

6. CONCLUSION

In this paper we have build upon the work of [20] and shown an algorithm based on FD-Graphs (a variant of directed hyper-graphs) that can be used to compute in polynomial time (i) the overall assurance indicator of a complex business process from its components for arbitrary monotone composition functions, (ii) the subpart of the business process that is responsible for such assurance indicator (i.e. the best security alternative). In contrast to standard hyper-graph algorithms [4] the propagation functions that we support must only be monotone. In this way we can capture a larger class of methods for security appraisals.

The most difficult point in the process is the concrete determination of propagation functions. We are planning to test several sets of propagation functions for various assurance indicators analyzing data from the corresponding case study in the SERENITY project. Once these functions are determined the proposed algorithm will assess all possible system configurations and if a security service is added/changed only the information about this service has to be updated. Other input parameters (weights and values of the leaf nodes) can be taken from business process specification (e.g. average time for execution of an activity) and from statistics or agreements with the partners.

The current activity is towards the creation of a tool to support security analysis. Another direction is to adapt the algorithms for dynamic changes [4, 3].

7. 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] G. Ausiello et. al. Graph algorithms for functional dependency manipulation. *JACM*, 30:752–766, 1983.

[3] G. Ausiello et. al. Directed hypergraphs: Problems, algorithmic results, and a novel decremental approach. In *Proc. of ICTCS'01*, 2001. Springer-Verlag.

[4] G. Ausiello et al. Optimal traversal of directed hypergraphs. Technical Report TR-92-073, International Computer Science Institute, Berkeley, CA, 1992.

[5] E. Bertino et. al. The specification and enforcement of authorization constraints in workflow management systems. *ACM TISSEC*, 2(1):65–104, 1999.

[6] S. A. Butler. Security attribute evaluation method: a cost-benefit approach. In *Proc. of ICSE'02*, 2002.

[7] V. Casola et. al. A SLA evaluation methodology in Service Oriented Architectures. In *Proc. of QoP'05*, 2005. Springer-Verlag.

[8] E. W. Dijkstra. A note on two problems in connexion with graphs. *NM*, 1:269–271, 1959.

[9] Eilifsen, Aa. et al. The Demand Attributes of Assurance Services and the Role of Independent Accountants. *Int. J. of Auditing*, 10: 143-162.

[10] G. Gallo, G. Longo, S. Pallottino, and S. Nguyen. Directed hypergraphs and applications. *Discr. App. Math.*, 42(2-3):177–201, 1993.

[11] R. Henning. Security service level agreements: quantifiable security for the enterprise? In *Proc. of NSPW'99*, pp. 54–60. 2000.

[12] Information Assurance Solutions Tech. Directors Information Assurance Technical Framework. (US) National Security Agency. v.3.1, September 2002.

[13] ISACA. CobiT. available via `www.isaca.org/cobit/`, 2007.

[14] E. Johansson and P. Johnson. Assessment of enterprise information security - an architecture theory diagram definition. In *Proc. of CSER'05*, 2005.

[15] M. H. Kang et. al. Access control mechanisms for inter-organizational workflow. In *Proc. of SACMAT'01*, pp. 66–74, ACM Press.

[16] Y. Karabulut et. al. Security and trust in it business outsourcing: a manifesto. In *Proc. of STM'08*, pages 47–58. ENTCS 179. Elsevier Science, 2006.

[17] G. Karjoth et. al. Service-oriented assurance Ű comprehensive security by explicit assurances. In *Proc. of QoP'05*, 2005. Springer-Verlag.

[18] W. List. The common criteria – good, bad or indifferent? *Inf. Sec. Tech. Rep.*, 2(1):19–23, 1997.

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

[20] F. Massacci and A. Yautsiukhin. Modelling of quality of protection in outsourced business processes. In *Proc. of IAS'07*. IEEE Press. To appear in 2007.

[21] G. R.-G. M.C. Jaeger and G. Mühl. Qos aggregation in web service compositions. In *Proc. of EEE'05*, 2005.

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

[23] R. Ortalo et. al. Experimenting with quantitative evaluation tools for monitoring operational security. *IEEE TSE*, 25(5):633–650, 1999.

[24] A. Rodrigues et. al. Towards an integration of security requirements into business process modeling. *Proc. of WOSIS'05*, pages 287–297, 2005. INSTICC PRESS.

[25] G. Stoneburner et. al. Risk management guide for information technology systems. Tech. Report 800-30, NIST, 2001.

[26] Y. Wang and P. K. Ray. Evaluation methodology for the security of e-finance systems. In *Proc. of EEE'05*. IEEE Computer Society Press, 2005.

[27] T. Yu and K.-J. Lin. A broker-based framework for qos-aware web service composition. In *Proc. of EEE'05*, pp. 22–29, 2005. IEEE Computer Society.