Information Technology

Privacy Analysis of User Behavior Using Alignments

M.Sc. Arya Adriansyah: Eindhoven University of Technology, P.O.Box, 513, 5600 MB Eindhoven, Netherlands
E-Mail: a.adriansyah@tue.nl
Arya Adriansyah is a PhD candidate in the Architecture of Information Systems group at Eindhoven University of Technology. He obtained his M.Sc. in Computer Science (2009) at the same university. His research interests include business process conformance checking, performance analysis, and visualization of research results.

Dr. Boudewijn F. van Dongen: Eindhoven University of Technology, P.O.Box, 513, 5600 MB Eindhoven, Netherlands
E-Mail: b.f.v.dongen@tue.nl
Boudewijn van Dongen is an assistant professor in the Information Systems group of Technology, Eindhoven. He received his Ph.D. in 2007, after successfully defending his thesis entitled “Process Mining and Verification”. His research interests range from process mining and process verification to supporting flexible processes and visualization of research results. Furthermore, he plays an important role in the development of the open-source process mining framework ProM, freely available from www.processmining.org.

Dr. Nicola Zannone: Eindhoven University of Technology, P.O.Box, 513, 5600 MB Eindhoven, Netherlands
Tel: +31-40-2472646, E-Mail: n.zannone@tue.nl
Nicola Zannone received the Ph.D. in Computer Science at the University of Trento in 2007. Since November 2012, he is assistant professor in the Security Group at Eindhoven University of Technology. His research interests include computer security, formal methods, privacy and data protection.

Keywords: Root causes analysis, privacy metrics, infringements assessment

Schlagworte: keywords in german

MS-ID: Heft: 53/* (2011)	n.zannone@tue.nl	September 23, 2013

Abstract

Privacy is becoming a urgent issue in information systems nowadays because of the stringent requirements imposed by data protection regulations. Traditional security approaches based on access control and authorization are not adequate to address these requirements. The underlying fundamental problem is that those approaches are preventive and thus they are not able to deal with exceptions. In this paper, we present a practical privacy framework that shifts the problem of preventing infringements into a problem of detecting infringements. The framework is based on systematic log auditing, use of patterns and privacy metrics to detect and quantify infringements.

Zusammenfassung

abstract in german

1 Introduction

Policy specification and enforcement have been largely adopted to guarantee that data are accessed and distributed according to established policies. However, preventive mechanisms do not eliminate privacy risks completely. For example, data re-purposing cannot be avoided using preventive means: a user may process the data for purposes other than those for which the data were originally accessed. To reduce privacy risks, we need flexible yet automated tools able to analyze user behavior, quantify the severity of infringements, and take compensation actions when the consequences of infringements cannot be tolerated.

Several compliance checking approaches (e.g., [10, 14]) have been proposed to analyze user behavior against specifications. Although these techniques are able to detect whether a deviation occurred, they do not explicitly identify the root causes of deviations. This makes it difficult to quantify the severity of deviations and, thus, to decide compensation actions.

The notion of alignments [17] provides a robust approach to determine the root causes of deviations. However, existing approaches for computing optimal alignments (e.g., [1, 2]) only show low level deviations, i.e. elementary deviations like insertions and suppressions. While low level deviations indicate where the process deviates, assessing the severity of infringements requires understanding how the process deviates [6]. Therefore, low level deviations needs to be combined into high level deviations like replacements and swappings of activities.

Identifying low level deviations and then using them to diagnose high level deviations may lead to misinterpretation of the root causes of deviations [4]. In addition, existing conformance techniques either consider every deviation equally or rely on manually predefined cost functions which define the cost of deviations. In the former, the quantification of infringements is meaningless from a privacy perspective while the latter implies that domain experts need to assign costs to all deviations manually, which is often time consuming.

The contribution of this paper is twofold. First, we extend alignment-based deviation analysis techniques to provide diagnostics on high-level deviations, hence providing a more accurate diagnosis of deviations than classical optimal alignments. In particular, we propose a pattern-based approach to explicitly identify high level deviations in the process model. Moreover, we present a metric to assess deviations from a privacy perspective. To this end, we identify a number of privacy factors and show how these factors can be used to assess infringements. To make the discussion more concrete, we demonstrate the approach using patterns to identify replacements and swappings.

2 Alignments

The intended system behavior is typically specified as process models. We consider process models in the form of classical Petri nets [11]. Petri nets consist of transitions, places, and directed arcs between them. Transitions are labeled with the tasks they represent. The state of a Petri net is represented by a multi-set of tokens on the places of the net, i.e. *the marking*. A transition is *enabled* if all its input places contain at least a token. When an enabled transition is executed or *fired*, a token is taken from each of its input places and a token is added to all output places. A sequence of transitions is a *complete firing sequence* of a net if firing transitions in the sequence from the initial marking of the net leads to its proper termination state, i.e. the *final marking*.

Fig. 1 shows a process of collecting diabetes patients' medical data for a trial in a hospital. First, a patient is contacted to make an appointment. If the patient agrees, he needs to sign a letter of consent. Then, a researcher inserts the patient's biodata through an online system (insert biodata). Afterward, the patient takes a general test, followed by a series of lab tests. In parallel, another researcher may examine his medical record. The process ends when the first researcher finalizes the process.

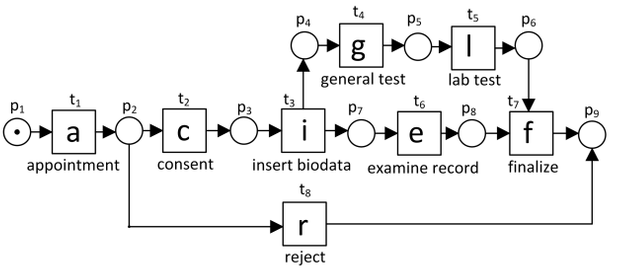

Fig. 1 A medical data collection process

$$\gamma_1 = \begin{array}{|c|c|c|}\hline \mathsf{a} & \mathsf{c} & \mathsf{r} \\ \hline \mathsf{a} & & \mathsf{r} \\ t_1 & \gg & t_8 \\ \hline \end{array} \quad \gamma_2 = \begin{array}{|c|c|c|c|c|c|c|c|}\hline \mathsf{a} & \mathsf{c} & \mathsf{r} & \gg & \gg & \gg & \gg & \gg \\ \hline \mathsf{a} & \mathsf{c} & & \mathsf{i} & \mathsf{g} & \mathsf{l} & \mathsf{e} & \mathsf{f} \\ t_1 & t_2 & \gg & t_3 & t_4 & t_5 & t_6 & t_7 \\ \hline \end{array}$$

Fig. 2 Alignments between the net in Fig. 1 and $\sigma_1 = \langle \mathsf{appointment}, \mathsf{consent}, \mathsf{reject} \rangle$

Task executions are often *logged* by information systems. Logged instance of a task is called an *event*. The sequence of events corresponding to a process instance is called *trace*. Given a trace and a Petri net, an *alignment* maps the trace to a complete firing sequence of the net (see [1] for a formal definition of alignment). Take for example a trace $\sigma_1 = \langle \mathsf{appointment}, \mathsf{consent}, \mathsf{reject} \rangle$. Fig. 2 shows two possible alignments between σ_1 and the net shown in Fig. 1, where tasks are abbreviated with their first letter. The top row of alignments shows the sequence of events in the trace; the bottom row shows the sequence of transitions in the net that yields a complete firing sequence along with the corresponding label. Deviations are explicitly shown by columns that contain $\gg$. For example, the second column in γ_1 shows that an

event occurs in the trace although it is not allowed according to the net, i.e. *move on log*. The fourth to eighth column in γ_2 show that some tasks must occur in σ_1 according to the net, but they are absent in the trace, i.e. *move on model*. Other columns for which events match the label of transitions represent *synchronous moves*.

As shown in Fig. 2, there can be more than one alignment between a trace and a Petri net. To determine the quality of alignments, a cost is assigned to each *movement* in the alignment. For instance, the *standard cost function* [17] assigns 1 to all moves on log/model and 0 for all synchronous moves. An *optimal alignment* between a trace and a Petri net according to a cost function is the one with the least total cost of deviations, i.e. if the standard cost function is used, it is the one with the least number of $\gg$.

In the remainder of this paper, we use the standard cost function unless indicated otherwise. In our example, γ_1 in Fig. 2 is an optimal alignment as it contains the least number of deviations (i.e., one deviation) among all possible alignments between σ_1 and the model in Fig. 1.

3 High-Level Deviations

This section presents an approach to detect high-level deviations together with guidelines for the definition of cost functions.

3.1 Patterns

Optimal alignments explicitly show low-level deviations in form of suppressed (i.e., moves on model) and inserted tasks (i.e., moves on log). However, to identify root cause of deviations, low-level deviations have to be analyzed and correlated. Take, for example, the net in Fig. 1 and trace $\sigma_2 = \langle$appointment, consent, lab test, general test, insert biodata, examine record, finalize$\rangle$. An optimal alignment between σ_2 and the net is shown in Fig. 3. In this example, the cause of deviation is a swap of task insert biodata with lab test. However, the optimal alignment in Fig. 3 shows two moves on log and two moves on model.

$\gamma_3 =$

a	c	l	$\gg$	g	i	e	$\gg$	f
a	c		i	g		e	l	f
t_1	t_2	$\gg$	t_3	t_4	$\gg$	t_6	t_5	t_7

Fig. 3 An optimal alignment between trace σ_2 and the net in Fig. 1, identifying only low-level deviations

To tackle this problem, we explicitly model high-level deviations in the process model. The general idea is shown in Fig. 4. High-level deviations are represented as Petri nets (i.e., *deviation patterns*), which are appended to the original Petri net. Then, optimal alignments are computed over the extended net. To ensure that optimal alignments between traces and appended nets do not contain any solution for which only some transitions of the pattern appear as synchronous moves in the alignment (i.e, either all or no transitions of the pattern are included), a very high cost (e.g., $+\infty$) is assigned to all moves on model for transitions in deviation patterns. The cost of the deviation is associated to first transition of the corresponding pattern.

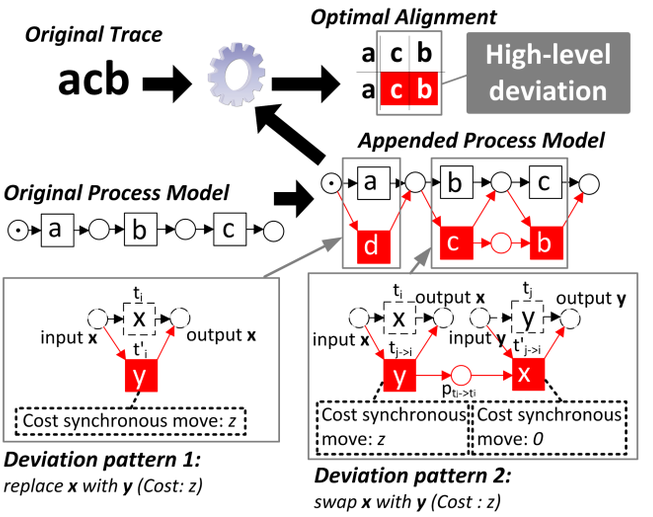

Fig. 4 Two sample deviation patterns and the general scheme to identify high-level deviations

For example, consider the *swapping* of task x with y as a possible high-level deviation with cost z (pattern 2 in Fig. 4). For all pairs of transitions (t_i, t_j) where t_i is labeled with x and t_j is labeled with y, we duplicate transitions t_i and t_j as $t_{j\to i}$ and $t'_{j\to i}$ respectively. Each duplicate has the same input and output places as the original transition. However, the label of duplicates are swapped, i.e. we assign to $t_{j\to i}$ the same label of t_j, and to $t'_{j\to i}$ the same label of t_i. To impose a ordering between the two duplicates, they are connected through a new place $p_{t_j\to t_i}$. This way, tokens can only exist in $p_{t_j\to t_i}$ if only $t_{j\to i}$ is fired, and $t'_{j\to i}$ is the only transition that can be fired. All moves on model involving the transitions in the appended pattern have cost $+\infty$; thus, all alignments with moves on model on these transitions are always discarded during the computation of optimal alignments. Synchronous moves involving the newly appended transitions has cost 0 except the first of the deviation pattern which has cost of the corresponding deviation.

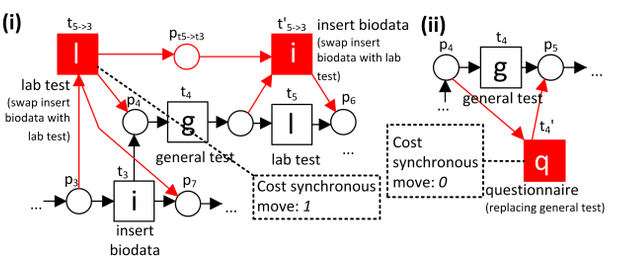

Fig. 5 Excerpt of the net in Fig. 1, appended

with deviation pattern for (i) swapping task insert biodata with lab test with cost 1, and (ii) replacing task general test with questionnaire with cost 0

Fig. 5(i) shows an excerpt of the net in Fig. 1 augmented with the deviation pattern representing the swapping of task insert biodata with lab test. Colored transitions and arcs in the figure form the deviation pattern. Suppose that the cost of swapping the two tasks is 1, Fig. 6 shows an optimal alignment between the appended Petri net and trace σ_2 where the total cost is 1. By pairing each synchronous move of $t_{i\to j}$ to its closest synchronous move of $t'_{i\to j}$, we obtain information about which tasks are swapped. In Fig. 6, the pair $(t_{5\to 3}, t'_{5\to 3})$ shows that insert biodata is swapped with lab test. Compared to the one shown in Fig. 3, the alignment in Fig. 6 clearly shows a better insights into the root cause of deviations.

$\gamma_4 =$

a	c	l	g	i	m	f
a	c	l	g	i	m	f
		(swap i with l)		(swap i with l)		
t_1	t_2	$t_{5\to 3}$	t_6	$t'_{5\to 3}$	t_6	t_7

Fig. 6 An optimal alignment between trace $\sigma_2 = \langle \mathbf{a}, \mathbf{c}, \mathbf{l}, \mathbf{g}, \mathbf{i}, \mathbf{m}, \mathbf{f} \rangle$ and the net in Fig. 5, showing that task insert biodata is swapped with lab test

A similar approach can be used to identify *replacements* of tasks. A task is replaced by another task in a trace if an instance of the latter yields the same effect as an instance of the former. Pattern 1 in Fig. 4 is a replacement pattern where all moves on model for colored transitions have cost $+\infty$. Consider again the net in Fig. 1. Suppose that in some trial, task general test can be replaced with questionnaire. To identify such a replacement, we append an instance of the pattern to the original net (see Fig. 5(ii)).

We showed that high-level deviations such as swappings and replacements can be identified by appending deviation patterns to the original process models. The same approach can be used to identify all high-level deviations that can be expressed in form of low-level deviations, such as swappings of sub-processes and replacements of sequences of tasks.

3.2 Cost Function

Existing conformance metrics only consider moves on model and moves on log to assess the conformance of a trace with a process model. Moreover, they usually consider the number of identified deviations to assess compliance with specification. This approach, however, is not suitable to analyze high-level deviations; also, it does not discriminate deviations with respect to their severity as the number of deviations does not necessarily reflect the overall severity of deviations. In this section, we present an approach for the definition of cost functions which addresses above drawbacks.

The method for quantifying deviations depends on whether the pattern can be observed from the execution of a single task (e.g., insertions, suppressions, and replacements) or whether observation of more than one task is required (e.g., swappings). The cost of deviations that can be observed from the execution of a single task requires comparing the task that has been executed with the task that should have been executed according to the specification (Note that insertions and suppressions can be seen as particular cases of replacement where the substituting task and the substituted task is the "empty task" respectively). However, not all replacements may have the same severity. For instance, replacing task general test with questionnaire can be considered less severe than replacing general test with physical examination. Therefore, to better quantify deviations we use the *degree of similarity* between the tasks that has been actually executed and the task in the specification. Intuitively, higher the similarity degree between the two tasks, lower is the cost of replacing one task with the other. Several existing methods (e.g., [9, 12, 13, 15]) can be used to compute the syntactic and semantic similarity between two terms. Here, we use ontology alignment techniques to compute the degree of semantic resemblance between two tasks.

The definition of the cost function for deviations that require the observation of more than one movement in the alignment depends on the type of deviation. For example, the cost of swappings can be determined by analyzing the tasks and control flow in the process model. The control flow defines a precise ordering in which tasks should be performed. In some case, when the swapping of two tasks has an insignificant or limited impact on the execution of the process model, the constraints imposed by the control flow can be relaxed. However, not every swapping may be allowable. In particular, a task cannot be swapped with another task if the execution of the latter is required for the execution of the former. We call such pairs of tasks *unswappable tasks*, and indicate with (a_1, a_2) that a_1 cannot be swapped with a_2.

The set of unswappable tasks specifies the basic pairs of tasks that cannot be swapped. The constraints imposed by this set may entail constraints on other pairs of tasks. For instance, a task should not be swapped with the predecessors of a task whose execution is necessary for the execution of the former activity. Indeed, allowing such a swapping will result in an undesirable alignment. Consider, for instance, the process in Fig. 1 and the pair of unswappable tasks (consent, insert biodata). If appointment is swapped with insert biodata, this would make it possible to perform insert biodata before consent.

To prevent the detection of misleading swapping, we introduce the *unswappable task closure*. Given a Petri net and a pair of unswappable tasks (a, b), i.e. a must not be swapped with b, the closure of (a, b) consists of

the set of pairs of tasks (a, c) such that c is a predecessor of b in the net and the set of pairs of tasks (d, b) such that d is a successor of a in the net. The costs of movements in the unswappable tasks closure is set to infinitely large $(+\infty)$. A pair of tasks can be swapped only if such a pair does not belong to the unswappable task closure. Its cost should be proportional to its impact on the process execution.

4 Quantifying Privacy

The cost function presented in the previous section only uses the task that has been executed to determine compliance. Although the executed task provides an indication of the purpose for which personal data have been used [14], this information alone is not sufficient to assess infringements from a privacy perspective. For instance, consider an execution of the net in Fig. 1 in which the medical record is examined by a receptionist rather than a researcher. If only the task is considered, the infringement will remain undetected.

To determine the severity of deviations from a privacy perspective, we extend the cost function in Section 3.2 by also considering the user who executed the task and the data accessed during its execution.[1]

Data protection regulations impose that personal data are collected for specified and lawful purposes and not processed in ways that are incompatible with their intended purposes [7]. Thus, *personal data* play a central role in privacy protection. Certain data items may be particularly sensitive for the data subject. There exist a number of qualitative [16] and quantitative [8] approaches which aim to quantify the sensitivity of personal data in order to regulate their disclosure. To quantify the amount of privacy loss caused by an infringement, we introduce the notion of *privacy weight*. Similarly to [8], privacy weights specify the cost of using a certain personal data item in the execution of the process. Intuitively, the higher the privacy weight of a data item, the less a user wishes to disclose that item.

The *role* by the user who executed the task is another factor for characterizing privacy [5]. A role describes job functions and responsibilities within an organization and is usually associated with the access rights necessary to achieve assigned duties. If a task is performed by a user that held a role different from the one defined in the specification, there is a risk of data misuse. However, not all the situations in which a task is executed by a user holding a role different from the prescribed one may present the same risk level. Similarly to the executed task, the degree of similarity between the role held by the user executing the task and the role defined in the specification should be considered to determine the severity of infringements (see Section 3.2).

As an example of cost function, we present how to calculate the severity of deviations that can be observed from a single movement in the alignment. Let e be an event in the log and a an activity which was supposed to be executed. The severity of the infringement $\Phi(a, e)$ can be calculated as follows:

$$\Phi(a, e) = \frac{1 + \Omega}{s_R \cdot s_T} - 1 \tag{1}$$

where $s_R \in [0, 1]$ is the degree of similarity between the role of the user executing the activity and the role associated to the activity in the specification (1 means that the two roles are semantically equivalent and 0 that they are completely incompatible); $s_T \in [0, 1]$ is the degree of similarity between the task which is actually executed and the task defined in the specification (1 means that the two tasks are semantically equivalent and 0 that they are completely incompatible); $\Omega \in \mathbb{R}^+$ represents the penalty due to unauthorized access to data. Penalty Ω is the sum of the privacy weights of the data items accessed during the actual execution of the task which were not supposed to be accessed according to the specification and were not already accessed by the user.

Consider the example in Fig. 7. The transitions in the net are annotated with a label which specifies a task, the role that the user executing the task should hold, and the data that should be accessed for the execution of the task. The transition in the pattern (red) indicates the deviation that may occur. In particular, the figure shows that the task may be executed by a doctor. Suppose that the similarity degree between researcher and doctor is 0.6. The cost of the deviation is 0.67.

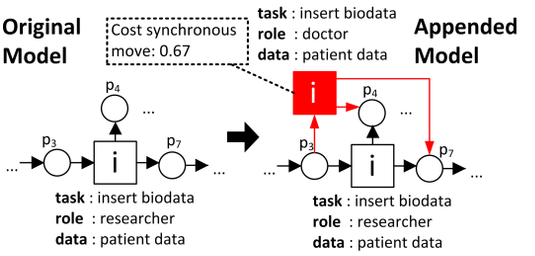

Fig. 7 Appended model to identify execution of insert biodata performed by a doctor

5 Prototype and Applications

The deviation identification approach proposed in this paper has been implemented as a part of the conformance analysis plug-in in the ProM 6 framework (see `http://www.processmining.org`). The plug-in aims to assist auditors in analyzing event logs and assess the severity of privacy deviations.

The plug-in takes as input a process model that describes the expected behavior, an event log that records

[1] Although other factors may be used to assess the severity of deviations, we focus on this information because it is usually available in event logs.

the actual behavior and a cost function. The plug-in constructs appended net and uses state exploration techniques [1, 2] to compute an optimal alignment between each trace in the log and the appended net with respect to the cost function. The computed alignment explicitly shows low-level deviations (i.e., suppressions, insertion) as well as high-level deviations in form of swapped and replaced activities. Fig. 8 shows the result of aligning trace ⟨appointment, consent, lab test, questionnaire, insert biodata, examine record, finalize⟩ with the net in Fig. 1 where general test is replaced with questionnaire and insert biodata is swapped with lab test. The overall severity of the trace is given by the sum of the severity of the two deviations. An auditor can use this diagnostics information to evaluate privacy violations and decide compensation actions.

Note that more than one optimal alignment may exist. In this case, all computed optimal alignments should be considered to obtain a complete insights into privacy violations. However, if only the severity of deviations is needed, considering one optimal alignment per trace is sufficient as all optimal alignments have the same cost.

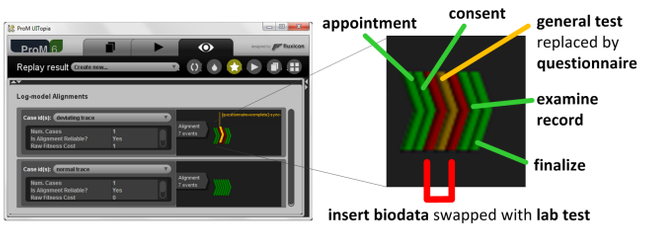

Figure 8 Screenshot of the plug-in, showing that high-level deviations are explicitly shown

6 Conclusions

This paper presented a novel framework for conformance checking that is able to detect high level deviations as well as to quantify deviations from a privacy perspective. The framework is based on the notion of alignments and makes use of privacy metrics to assess the severity of deviations. We performed a number of experiments that show that the proposed framework is robust in the sense that it can detect and quantify deviations without a prior knowledge about deviations [3].

Acknowledgments. This work has been funded by the Dutch program COMMIT under the THeCS project, the Cyber Security program under the PriCE project, and the EU Commission under the Seventh Framework Project "PoSecCo".

References

[1] A. Adriansyah, N. Sidorova, and B. F. van Dongen. Cost-Based Fitness in Conformance Checking. In *Int. Conf. on Application of Concurrency to System Design*, pages 57–66. IEEE, 2011.

[2] A. Adriansyah, B. van Dongen, and W. van der Aalst. Memory-Efficient Alignment of Observed and Modeled Behavior. BPM Center Report BPM-03-03, BPMcenter.org, 2013.

[3] A. Adriansyah, B. F. van Dongen, and N. Zannone. Controlling Break-The-Glass Through Alignment. *ASE SCIENCE Journal*, 2(4):198–212, 2013.

[4] S. Banescu, M. Petkovic, and N. Zannone. Measuring privacy compliance using fitness metrics. In *Business Process Management*, LNCS 7481, pages 114–119. Springer, 2012.

[5] S. Banescu and N. Zannone. Measuring privacy compliance with process specifications. In *3rd International Workshop on Security Measurements and Metrics*, pages 41–50. IEEE, 2011.

[6] B. Depaire, J. Swinnen, Mieke, and K. Vanhoof. A Process Deviation Analysis Framework. In *Business Process Management Workshops*, LNBIP 132, pages 701–706. Springer, 2013.

[7] P. Guarda and N. Zannone. Towards the development of privacy-aware systems. *Inf. Softw. Technol.*, 51(2):337–350, 2009.

[8] F. Massacci, J. Mylopoulos, and N. Zannone. Hierarchical hippocratic databases with minimal disclosure for virtual organizations. *VLDB J.*, 15(4):370–387, 2006.

[9] G. A. Miller. WordNet: a lexical database for English. *Commun. ACM*, 38(11):39–41, 1995.

[10] M. Montali, M. Pesic, W. M. P. van der Aalst, F. Chesani, P. Mello, and S. Storari. Declarative specification and verification of service choreographiess. *TWEB*, 4(1):3:1–3:62, 2010.

[11] T. Murata. Petri nets: Properties, analysis and applications. *Proc. IEEE*, 77(4):541–580, 2002.

[12] L. D. Ngan, T. M. Hang, and A. Goh. Semantic Similarity between Concepts from Different OWL Ontologies. In *Int. Conf. on Industrial Informatics*, pages 618–623. IEEE, 2006.

[13] H. Nguyen and H. Al-Mubaid. A Combination-based Semantic Similarity Measure using Multiple Information Sources. In *Int. Conf. on Information Reuse and Integration*, pages 617–621. IEEE, 2006.

[14] M. Petković, D. Prandi, and N. Zannone. Purpose control: Did you process the data for the intended purpose? In *Secure Data Management*, LNCS 6933, pages 145–168. Springer, 2011.

[15] D. Trivellato, F. Spiessens, N. Zannone, and S. Etalle. Reputation-based ontology alignment for autonomy and interoperability in distributed access control. In *Int. Conf. on Computational Science and Engineering*, pages 252–258. IEEE, 2009.

[16] A. Tumer, A. Dogac, and I. H. Toroslu. A semantic-based user privacy protection framework for web services. In *Intelligent Techniques for Web Personalization*, LNCS 3169, pages 289–305. Springer, 2003.

[17] W. M. P. van der Aalst, A. Adriansyah, and B. F. van Dongen. Replaying History on Process Models for Conformance Checking and Performance Analysis. *WIREs Data Mining and Knowledge Discovery*, 2(2):182–192, 2012.

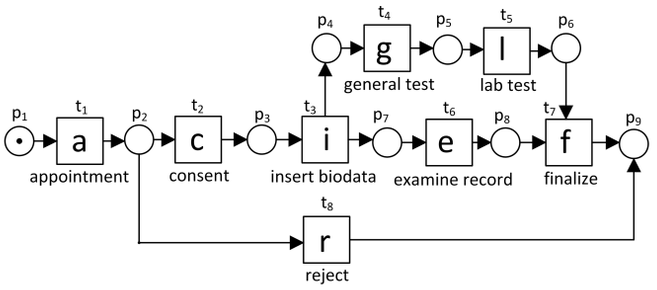

Figure 1: A medical data collection process

$$\gamma_1 = \begin{array}{|c|c|c|}\hline a & c & r \\ \hline a & & r \\ t_1 & \gg & t_8 \\ \hline \end{array} \quad \gamma_2 = \begin{array}{|c|c|c|c|c|c|c|c|}\hline a & c & r & \gg & \gg & \gg & \gg & \gg \\ \hline a & c & & i & g & l & e & f \\ t_1 & t_2 & \gg & t_3 & t_4 & t_5 & t_6 & t_7 \\ \hline \end{array}$$

Figure 2: Alignments between the net in Fig. 1 and $\sigma_1 = \langle \mathsf{appointment}, \mathsf{consent}, \mathsf{reject} \rangle$

$$\gamma_3 = \begin{array}{|c|c|c|c|c|c|c|c|c|}\hline a & c & l & \gg & g & i & e & \gg & f \\ \hline a & c & & i & g & & e & l & f \\ t_1 & t_2 & \gg & t_3 & t_4 & \gg & t_6 & t_5 & t_7 \\ \hline \end{array}$$

Figure 3: An optimal alignment between trace σ_2 and the net in Fig. 1, identifying only low-level deviations

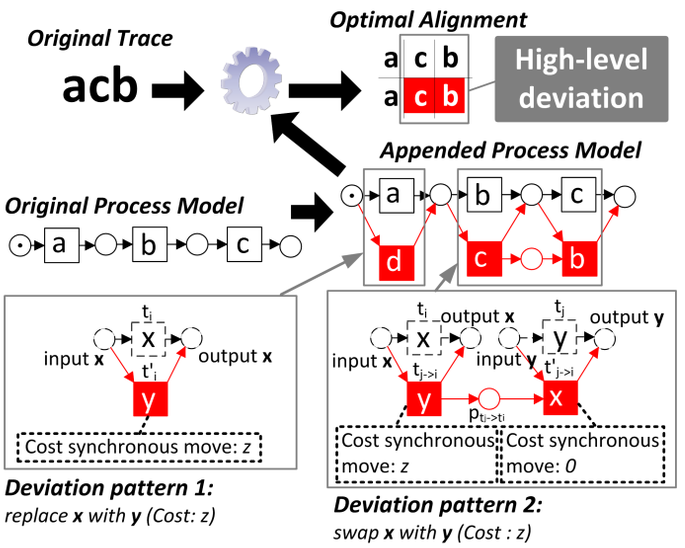

Figure 4: Two sample deviation patterns and the general scheme to identify high-level deviations

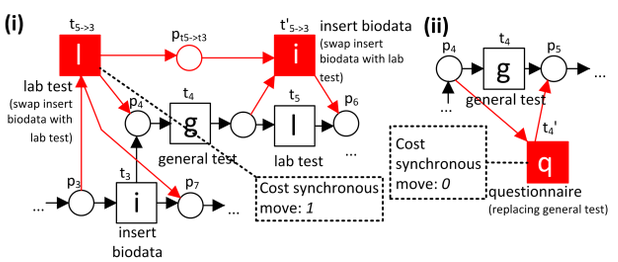

Figure 5: Excerpt of the net in Fig. 1, appended with deviation pattern for (i) swapping task insert biodata with lab test with cost 1, and (ii) replacing task general test with questionnaire with cost 0

$$\gamma_4 = \begin{array}{|c|c|c|c|c|c|c|}\hline \mathsf{a} & \mathsf{c} & \mathsf{l} & \mathsf{g} & \mathsf{i} & \mathsf{m} & \mathsf{f} \\ \hline \mathsf{a} & \mathsf{c} & \mathsf{l} & \mathsf{g} & \mathsf{i} & \mathsf{m} & \mathsf{f} \\ & & \text{(swap i with l)} & & \text{(swap i with l)} & & \\ t_1 & t_2 & t_{5\to 3} & t_6 & t'_{5\to 3} & t_6 & t_7 \\ \hline \end{array}$$

Figure 6: An optimal alignment between trace $\sigma_2 = \langle \mathsf{a}, \mathsf{c}, \mathsf{l}, \mathsf{g}, \mathsf{i}, \mathsf{m}, \mathsf{f} \rangle$ and the net in Fig. 5, showing that task insert biodata is swapped with lab test

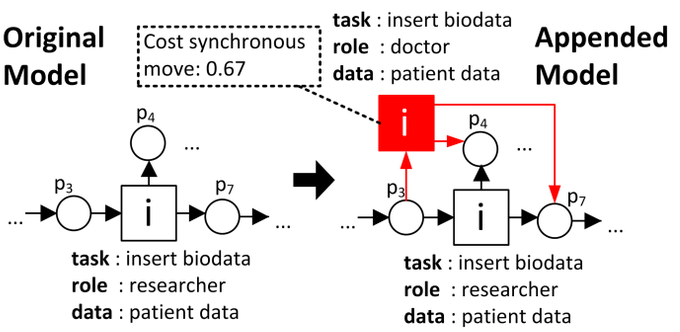

Figure 7: Appended model to identify execution of insert biodata performed by a doctor

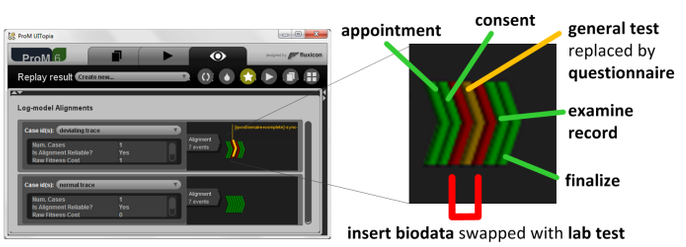

Figure 8: Screenshot of the plug-in, showing that high-level deviations are explicitly shown