"BPELanon": Anonymizing BPEL Processes

Marigianna Skouradaki[1], Dieter Roller[1], Cesare Pautasso[2], and Frank Leymann[1]

[1] Institute of Architecture of Application Systems, University of Stuttgart, Germany
{skouradaki,dieter.h.roller,leymann}@iaas.uni-stuttgart.de
[2] Faculty of Informatics, University of Lugano, Switzerland
c.pautasso@ieee.org

Abstract We are currently developing a performance benchmark for Workflow Management System. As a first activity we are collecting real-world processes. However, to protect their competitive advantage, some companies are not willing to share their corporate assets. This work's objective is to propose a method ("BPELanon") for BPEL process anonymization in order to deal with the problem. The method transforms a process to preserve its original structure and runtime behavior, while completely anonymizing its business semantics. Anonymization is a complicated task that must meet the requirements we outline in this paper. Namely, we need to preserve the structural and executional information while anonymizing information such as namespaces, names (activity names, variable names, partner link names etc.), and XPath expressions that may reveal proprietary information. Furthermore, the names contained in the anonymized process should be chosen carefully in order to avoid conflicts, preserve privacy, and file-readability. Multiple dependency relations among process artifacts raise the challenge of fulfilling the aforementioned requirements, as a unique change in a file potentially leads to a flow of changes to other related process artifacts.

Keywords: Anonymization, BPEL, Workflows, Business Processes

1 Introduction

Given the fact that "process equals product" [3] most companies and business organizations are not willing to share their process models with academic researchers due to competitive reasons to protect their intellectual property. Since our first goal with the "BenchFlow" project[1] is to collect real-world business process models that can be later used to synthesize a Benchmark, we want to encourage sharing of models that are suitable for our purposes without revealing critical company information. The contributions of this work are as follows:

1. identify the requirements of anonymization methodology
2. propose a method ("BPELanon") that exports the anonymized process model containing the original BPEL process without its business semantics, but solely its executable structure

[1] http://www.iaas.uni-stuttgart.de/forschung/projects/benchflowE.php

N. Herzberg, M. Kunze (eds.): Proceedings of the 6th Central European Workshop on Services and their Composition (ZEUS 2014), Potsdam, Germany, 27-03-2014, published at http://ceur-ws.org

2 Approaching the Problem

2.1 Requirements

The design of "BPELanon" must address the following initial list of requirements identified during our work in various research projects, and especially during our collaboration with industry partners: The main requirement and purpose of methodology is to:

R1: Support both pseudonimization and anonymization of data upon the user's choice. Pseudonimization is the technique of masking the data, while maintaining ways to the original data [1]. On the contrary, anonymization changes the critical data and makes it impossible to trace back the original version of data [4]. Providing the option of pseudonimization makes it possible for the originator to trace bugs or inconsistencies found in the anonymized file, and apply changes to the original process.

In order to satisfy [R1] a number of other requirements occur. These can be grouped to requirements that stem from the XML nature of BPEL:

R2: Scramble the company's sensitive information that can be revealed in activity names, variable names, partner link names, partnerlink type names, port type names, message names, operation names, role names, XSD Element names, namespaces, and XPath expressions. The name choice for these attributes is usually descriptive, and reflects the actual actions to which they correspond. So they can reveal a lot of the process semantics.
R3: The exported process model must not contain namespace information in incoming links to external web sites that reveal business information (backlinks)
R4: The exported process model must not contain names (including activity names, variable names, partner link names, partnerlink type names, message names, operation names, role names, and XSD Element names) with backlinks to business information
R5: The exported process model must not contain XPath expressions with backlinks to business information. If no custom XPath functions are used, [R5] is a consequence of requirement [R4].
R6: Remove description containers (comments and documentation) that reveal critical information and semantics.

BPEL-specific requirements:

R7: The exported process model must keep the structural information and executability
R8: The exported process must maintain an equivalent run-time behavior
R9: The exported process must maintain equivalent timing behavior

The following requirements are related to the renaming methodology that will be applied:

R10: It has to be ensured that the scrambled name prevents reverse engineering to get the original names. For example if data is encrypted with a known function (e.g. RSA, MD5) and we know the used key, then it is easy to obtain the original data.

R11: Names must be chosen in a way that conflicts are avoided between the original and exported file. For example an easy name choice would be to change each name with respect to its type followed by an ascending ID. For example the name of activity "Payment" could have been changed to the name "Activity1". Nevertheless, this way is not considered safe. "Activity1" could also have been a possible name choice for the original process model as it is a word frequently met in Business Process Management. This would lead to a sequence of conflicts. Which elements named "Activity1" correspond to the anonymized element and which to the one contained in the original process?

R12: The names must lead to an human-readable exported file. For example let's assume that we use UUIDs for name choice. That would lead to activity names such as: `f81d4fae-7dec-11d0-a765-00a0c91e6bf6`. The exported file will not be easy to read for humans.

2.2 Challenges

This section analyzes the challenges that stem from the need to satisfy the requirements described in

2.3 Requirements

Each process specification is wrapped in a package which is a directory containing all deployment artifacts. At the minimum the directory should contain a deployment descriptor, and one or more process definitions (BPEL), WSDL, and XSD files[1]. Many dependency relations among files as shown in Fig. 1 increase the complexity of anonymization as small changes in a file may lead to numerous subsequent changes to other process artifacts [Challenge 1]. The complexity increased by the need to meet Requirement 2 [Challenge 2]. The renaming methodology also needs to be carefully examined in order to satisfy Requirements 9–12 [Challenge 3].

The BPEL-specific requirements reveal a new set of challenges that will be more complex to fulfill. How do data and data specific decisions affect the runtime behavior of the anonymized model? [Challenge 4]. How is BPEL lifecycle affected by anonymization? [Challenge 5]. To what extend will timing behavior be maintained? [Challenge 6]. These challenges will be addressed in future work.

Following the approach of "divide and conquer" the anonymization methodology followed for each artifact should be separately and carefully examined. In this paper we focus on the BPEL - WSDL anonymization aiming to satisfy [Challenges 1, 2, 3].

[1] `http://ode.apache.org/creating-a-process.html`

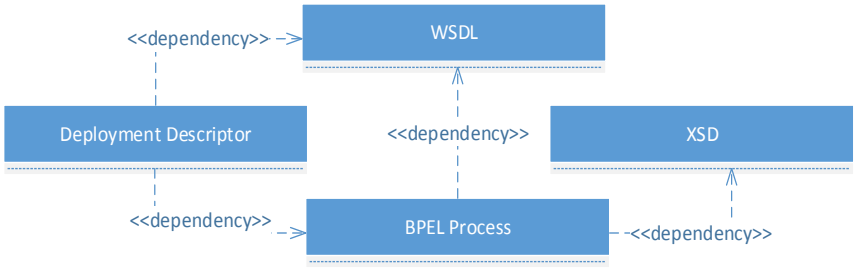

Figure 1. Dependency relations among artifacts to be anonymized

Fig. 2 shows a more detailed analysis of the occurring dependencies between the BPEL and WSDL artifacts. The grey entities represent the BPEL elements while the green entities represent WSDL elements. The directed associations that connect the members with each other show dependency between the entities. The arrow shows the "direction" of dependency. This means that the member to which the arrow leads is an artifact which creates high dependencies between the rest of the participating entities. Therefore when this member is changed the interconnected members should be accessed and changed correspondingly.

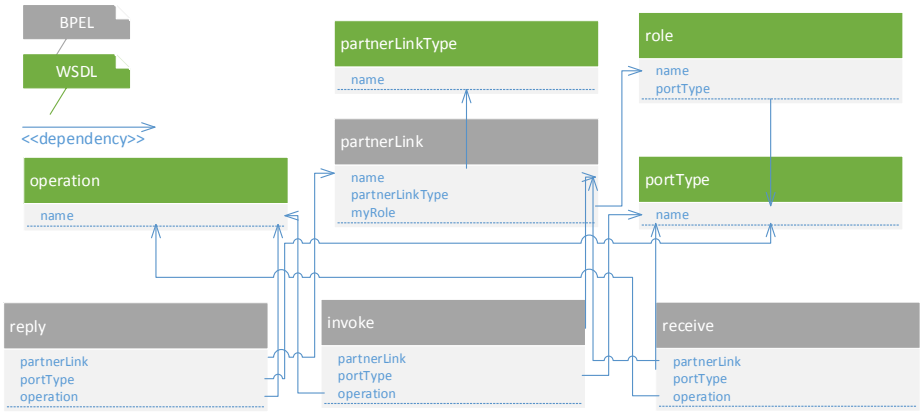

Figure 2. Dependencies between BPEL and WSDL files of a Business Process

3 Designing the Method

This section describes the methodology that is used for developing "BPELanon". Elements in a BPEL file can be divided into three groups:

– Free Elements Group: Elements that need to be anonymized, but are not bound to changes that occurred in other files.

- WSDL Bounded Group: Elements that need to be changed because they were bounded with elements that are changed in the WSDL file.
- Internally Bounded Group: Elements that need to be changed because they are bounded to other changed elements within the same file. Internally Bounded Groups can be found in both BPEL and WSDL files.

The anonymization of "Free Elements Group" is trivial. However, the anonymization of "WSDL Bounded Group" and "Internally Bounded Group" are more complex tasks. For its implementation we need a "Registry of Alterations". This is a registry of metadata that is created during the anonymization a file and logs the occurring changes. It must contain in the minimum the following information: the element's type, and the corresponding attributes' new and old data.

The main idea of the anonymization is to scan the documents (WSDL, BPEL does not matter) looking for element attributes that might contain semantics (critical attributes) and need to be scrambled, and adding them to the "Registry of Alterations" the old and new value. The information on which attributes are critical can be stored with metadata. Next we scan the documents looking for references to the scrambled elements and update their values. Below it is described the anonymization method for the "WSDL Bounded Group".

Anonymization starts with the creation of a metadata schema that reflects the interconnections shown in Fig. 2. Next we construct a "Table of References" that shows correlation of a BPEL process and its WSDL files. This is done by parsing the `<bpel:import>` annotations of the BPEL file. We then process the WSDL files, which contain the definitions for the artifacts that are referenced in BPEL. We run through each one of the WSDL files in "Table of References" and start anonymizing the attributes of the elements step by step. In order to fulfill [R8] the function of anonymization will pick random worlds of an English Dictionary [1]. As argued before a world of a well known human language will lead to more readable results than UUIDs. We only focus on the anonymization of critical attributes as not every attribute needs to be anonymized. By maintaining a "Registry of Alterations", we apply the subsequent changes to the BPEL. We have created an outer loop that repeats this process for each WSDL file. Another option would have been to parse all WSDL files and finally apply the changes to BPEL file in one parse. However WSDL files might have common names and this would lead to more complex solution. We have therefore chosen this safer although most likely more time consuming method.

At the end of the process "Table of References" and "Registry of Alterations" is destroyed if the tool is set to anonymize and not pseudonimize. The above procedure describes Algorithm 1. For the anonymization of the "Internally Bounded Group" a similar process needs to be followed.

4 Related Work

Attempts for anonymization can be found in various fields of computer science such as network security (filtering, replacement, reduction of accuracy etc. [6]) and

[1] `http://www.winedt.org/Dict/`

Algorithm 1 Anonymization process of BPEL-WSDL for "WSDL Bounded Group"

```
create TableOfReferences by parsing <bpel:import> annotations of BPEL
for all WSDL files W in tableOfReferences do
  for all elements E in W do
    a ← getCriticalAttributes(E)
    for all a do
      updateRegistryOfAlterations(E.type,a.type,a.data,"old")
      applyAnonymizationPattern(a.data)
      updateRegistryOfAlterations(E.type,a.type,a.data,"new")
    end for
  end for
  for all element E in BPEL file do
    a ← getCriticalAttributes(E)
    for all a do
      resultType ← findTypeOfInterconnection(E.type,a.type)
      a.data ← getNewValueOfAttribute(resultType,a.data) {from registryOfAl-
      terations}
    end for
  end for
end for
if anonymization then
  delete tableOfReferences
  delete registryOfAlterations
end if
```

database systems (data generation, encryption etc. [5], k-anonymity, l-diversity, and t-closeness[1]). These approaches cannot be applied to BPEL as they are tightly tailored to the architecture and principles of different technologies.

The tools XMLAnonymizer[2] and XMLAnonymizerBean[3] were found. XMLAnonymizer is a primary approach to anonymization that focuses on changing the attribute value of the XML file ([R4] partially covered). The XMLAnonymizerBean anonymizes elements and attributes by removing the namespaces of an XML file ([R3] partially covered). Overall, these utilities partially satisfy the requirements of "BPELanon". The "BPELanon" method is a more complex approach since it deals with all the requirements and challenges described in Sect. 2.

5 Conclusions and Future Work

In this paper we have proposed a method for the anonymization of BPEL processes. We focus on BPEL processes without extensions as experience shows

[1] http://arx.deidentifier.org/

[2] https://code.google.com/p/xmlanonymizer/

[3] http://help.sap.com/saphelp_nw04/helpdata/en/45/d169186a29570ae10000000a114a6b/content.htm

that BPEL is used widely in industry to implement workflows. There are more than 60 BPEL extensions available [2], but the processes we collected so far indicate that none of these extensions is used in real-world settings. We have analyzed a set of requirements and challenges that make process anonymization difficult. To address the requirements and challenges we suggest an algorithm that is a first approach to the methodology of business process anonymization. The main contribution of this paper is the design of a methodology with focus on BPEL anonymization.

In future work we will investigate what is the impact of anonymization to the BPEL process lifecycle, the ways that data and data dependent decisions are influenced by anonymization, and include timing behavior information into BPELanon methodology. The implementation of "BPELanon" has started, and will be tested with a set of workflows with various characteristics. The first release will be then distributed to companies for evaluation and usage. We intend to extend "BPELanon" in order to provide various options of anonymization, and anonymization valid for other languages. After collecting a sizable sample of anonymous process models, we will work on a method for "Statistical Analysis" that aims to calculate useful statistical information out of the BPEL process collection.

Acknowledgments This work is funded by the "BenchFlow" (LE 2275/7-1) project supported by German Research Foundation (DFG).

References

1. Federal Ministry of Justice: German Federal Data Protection Law (1990)
2. Kopp, O., Görlach, K., Karastoyanova, D., Leymann, F., Reiter, M., Schumm, D., Sonntag, M., Strauch, S., Unger, T., Wieland, M., Khalaf, R.: A Classification of BPEL Extensions. JSI 2(4), 2–28 (November 2011)
3. Leymannn, F.: Managing business processes via workflow technology. Tutorial at VLDB 2001 (11-14 September 2001)
4. Strauch, S., Breitenbücher, U., Kopp, O., Leymann, F., Unger, T.: Cloud Data Patterns for Confidentiality. In: Proceedings of the 2nd International Conference on Cloud Computing and Service Science, CLOSER 2012. pp. 387–394. SciTePress (2012)
5. Vinogradov, S., Pastsyak, A.: Evaluation of data anonymization tools. In: DBKDA 2012, The Fourth International Conference on Advances in Databases, Knowledge, and Data Applications. pp. 163–168 (2012)
6. Yurcik, W., Woolam, C., Hellings, G., Khan, L., Thuraisingham, B.M.: Toward trusted sharing of network packet traces using anonymization: Single-field privacy/analysis tradeoffs. CoRR abs/0710.3979 (2007)