Using Semantically Annotated Models for Supporting Business Process Benchmarking

Hans-Georg Fill

Stanford University BMIR / University of Vienna DKE,
251 Campus Drive, Stanford CA 94305, USA
fill@dke.univie.ac.at

Abstract. In this paper we describe an approach for using semantic annotations of process models to support business process benchmarking. We show how semantic annotations can support the preparation of process benchmarking data by adding machine-processable semantic information to existing process models without modifying the original modeling language, conduct semantic analyses for the purpose of performance measurement, and obfuscate the information contained in the models for ensuring confidentiality. The approach has been implemented on the ADOxx platform and applied to two use cases for a first evaluation.

Key words: Semantic Annotation, Benchmarking, Performance Evaluation, Ontologies

1 Introduction

Evaluating the performance of business processes and comparing it to internal and external benchmarks is an essential aspect in performance management [1]. The core feature of such benchmarks is to learn from others and then adapt one's own processes in order to gain competitiveness [2]. In this way, large, global companies aim today for the spreading and homogenization of their internal best practices across their units [3]. However, as processes are today also viewed as assets that represent the actual know-how platform of an organization and are thus essential for the creation of competitive advantage [4, 5], benchmarking should also allow for the combination with individual, innovative solutions [6].

As a basis for business process benchmarking it can be reverted to graphical modeling languages [7]. These can act as a foundation for process benchmarking by providing a machine-processable representation that can be easily shared between interested parties. However, as benchmarking may require additional or different information than was originally conceived when creating the models, additional manual effort is needed to prepare suitable models [8]. This concerns especially the requirements of comparability and confidentiality as well as the content-based analysis of the models.

To support users in preparing such data for analyses and benchmarking, we will describe an approach that builds upon semantically annotated process models. Thereby, the preparation and analysis of the process data is supported in the following ways: Through annotations with terms from a shared vocabulary the comparability of individually created process models can be ensured. In addition, we will show how semantic annotations enable ex-post analyses of the semantic content of the process models to semi-automate complex analysis tasks. Furthermore, the annotations create the basis for an obfuscation mechanism that deals with sensitive information and acts as an additional incentive to share the data. The remainder of the paper is structured as follows: In section 2 we will outline the foundations for our approach which will allow us to describe the approach in section 3. Section 4 shows the implementation and application to two uses cases, section 5 discusses the relations to existing approaches and section 6 will conclude the paper with an outlook on the future steps.

2 Foundations

In this section we will give a brief outline of the foundations for our approach. In particular, we will describe the aspects of business process benchmarking, the role of conceptual modeling for supporting benchmarking, and the characteristics of semantic annotations of conceptual models that are relevant for our approach.

2.1 Business Process Benchmarking

Benchmarking in general refers to the continuous measurement and comparison of an organization against other organizations, in particular business leaders, to get information helping to improve its performance [2, 1, 8]. Thereby, it is aimed at establishing objective measures of an organization's performance, the adaptation of best practices, and the incentive for introducing new and innovative concepts [6]. Traditionally, a distinction has been made between internal benchmarking, competitive benchmarking, and generic benchmarking [2, 9]. Internal benchmarking is particularly used in large organizations and characterizes comparisons against other units in the same organization. In competitive benchmarking the own performance of an organization is compared to its direct competitors and in generic benchmarking the comparison is performed regardless of industry. Especially in competitive benchmarking, confidentiality and sensitivity of data and information may pose potential problems. This concerns in particular *business process benchmarking*, which measures, compares, and exchanges the practices and ways of performing and not only the pure levels of performance [2]. Business process benchmarking is therefore able to give deeper insights into the capability and choices for improving an organization's performance.

Typically, business process benchmarking is comprised of four steps: The planning of the benchmarking project, the collection of process data, the analysis of the data for results, and the adaptation for improvement [1]. In the following we will focus on the collection of process data and the analysis of the data. In this way, at first the business processes to be benchmarked are selected and key process performance indicators are determined. Due to the large number of existing metrics we refer to a recent survey by Heinrich and Paech who compiled quality characteristics for process activities, actors, and inputs and outputs [10]. The comparison of these indicators then shows how much process improvement is possible in relation to the benchmarking partner, whereas the comparison of qualitative process information, i.e. the process flow, gives detailed insights on where and how improvements can be achieved [9].

2.2 Conceptual Process Models for Benchmarking

As mentioned in the introduction, organizations today often represent the knowledge about their business processes in the form of conceptual graphical models [7]. These are based on a modeling language with a formal syntax and a graphical notation that can serve as a basis for the implementation of according model editors [11]. One particular aspect - which we will show is of major importance for a machine-based support of benchmarking - is that the semantics of the labels in conceptual models are not formally defined but given in natural language [12]. This stems from the goal of conceptual modeling to support human communication and understanding, which does not require a formal representation of the meaning [13]. Although this greatly eases the understanding and handling of these types of modeling languages and has contributed to their widespread use [14], it also limits the options for processing the contained information with algorithms.

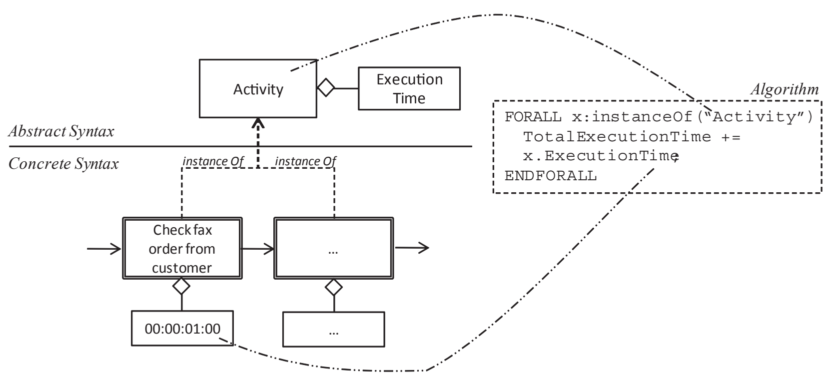


Fig. 1. Illustration of an Algorithm based on the Abstract and Concrete Syntax of a Business Process Model

For illustrating this with an example consider an abstract syntax element with the label "activity" (see also figure 1). To correctly interpret this element a semantic description has to be supplied for it. Thereby, it can be defined that elements of this type consume time, whose quantity is specified by the attached attribute "execution time". Among the many ways of expressing such semantic descriptions, one approach is to create an algorithm that explicitly references elements of the abstract syntax and can thus also process their instantiations. In this way semantic definitions may be specified independently of the instantiations of the element as they are targeted towards the abstract syntax that is the same for all resulting models.

However, when an element is instantiated, a user typically adds a label and a description to the element. Following the example, a label for the activity element could be "Check fax order from customer", together with a textual description that explains in detail which aspects should be checked. Without further specification, the semantic information contained in these natural language descriptions cannot be directly processed by an algorithm as no formal interpretation has been defined. In contrast to the specification of formal semantics for the abstract syntax, this semantic information entirely depends on the choice of the user at the time of creating a model and is not a-priori known.

When applying these considerations to benchmarking, an essential aspect is the computation of selected key performance indicators based on the process data. In particular, several types of indicators can already be derived by accessing the structural properties of the process models via their abstract syntax. This includes static metrics such as the number of events, activities or decisions as well as flow metrics such as the number of loops, parallel paths, joins or splits [15]. However, a large part of quality metrics require the interpretation of the content of the business processes. Examples for such metrics include but are not limited to the number of media disruptions, the fault density, the degree of automation or the effectiveness of documentation [10].

Depending on the type of the used modeling language, this information may however not be at all or only partly expressible by the abstract syntax. It may be at best added by the user through the labels in natural language at the time of modeling. And even more, it could also be the case that the labels alone do not reflect this information at all, but that it actually has to be added by the user specifically for the purpose of benchmarking. When aiming for an algorithmic support of benchmarking, it is though necessary to lift this information to a concept-based level where this information is provided in a machine-processable format.

2.3 Semantic Annotation of Conceptual Process Models

In order to explicate such semantic information it can be chosen from several directions. One is to create a *new modeling language*, either from scratch or by extending an existing one. This allows defining or extending the abstract syntax in a way that the parts of the semantic information, which are necessary for a particular algorithm can be expressed. It also implies however, that the modeling language and according models can be adapted in this way. Typically, this is accomplished by introducing new elements or attributes that provide the necessary information. Thereby, each instance can be assigned the semantic information required for running the algorithms. Examples for this direction are domain specific languages in general, the semantic building block-based language [16] and the visual templates described by [17]. Although such approaches provide many advantages in terms of machine-processing, their major drawback is that the modeler may not be able to represent all relevant world facts but is limited to the terms and concepts available in the extended abstract syntax [16].

Another direction is to use an *existing modeling language* and ensure the processability either by enforcing modeling conventions or by using references to machine-processable semantic schemata, i.e. *semantic annotations*. Modeling conventions may either be defined on an organizational level [18], e.g. by providing rules which terms can be used in a model. Or, they may be enforced automatically, e.g. by applying natural language processing techniques together with a domain vocabulary [19]. The use of semantic annotations has been described by several authors. Thereby, the references or meta-data can be expressed using different types of languages, ranging from conceptual languages that are based on natural language semantics to programming languages and logic based languages with formal semantic specifications. This direction has been successfully applied to several common business process modeling languages such as: Event-driven process chains [20, 21], the business process modeling notation [22] or Petri nets [23]. For these approaches the required semantic information is either derived by automatically looking up terms in a schema or linking them to the schema manually and then building on this information for further processing. In the following we will use this last direction for illustrating how semantic annotations can be used for supporting the data preparation and analysis step in business process benchmarking.

3 Model-based Benchmarking Using Semantic Annotations

To discuss our approach we will revert to meta models for describing the abstract syntax of the underlying modeling languages [11]. This will permit us to illustrate the relationships between business process models and semantic schemata in an intuitive way. Furthermore, by using meta models it will be possible to directly implement the modeling language on a corresponding meta modeling platform as well as specify algorithms for computing process performance indicators.

3.1 Definition of the Meta Model

We first define a meta model for describing the abstract syntax of a simplified business process model type (see figure 2). The meta model comprises elements for describing the control flow and information flow in business processes as well as the organizational structure. In contrast to real-world business process modeling languages both model types only contain a subset of possible modeling elements and relations. However, for the purposes of illustrating the concepts necessary for our approach this is sufficient and could easily be extended at any time. Similarly, also the attributes assigned to the model elements have been limited to the "name" attribute for process elements, the "probability" attribute for the sequence flow relation and the "execution time" attribute for the "activity" elements.

For describing machine-processable semantic schemata, we added elements to the meta model for representing concepts of the web ontology language (OWL). OWL has been chosen as it is currently widely used for representing formal semantic information. In particular, OWL builds upon description logics [24] and comes with a formal semantic specification. Thereby, it supports automated reasoning techniques for checking the consistency of an ontology and the entailment relationships between its concepts based on a set of axioms [25]. So it can for example be ensured by a reasoner that the semantic information stays consistent with previously defined concepts based on a detailed set of restrictions. Furthermore, OWL ontologies can be exchanged using an RDF/XML syntax and thus may be easily shared between different tools and platforms. In figure 2 the main elements of OWL are shown. To define semantic annotations of conceptual models, two options have been made available in the meta model. The first is by using the reference attribute "semantic annotation" that has been added to the "Process Element" and the "Organizational Element" super classes.

By using this attribute, semantic annotations can be easily expressed that result in a direct assignment of ontology concepts to instances of the concrete syntax in the business process and working environment models. However, for this kind of annotation it is necessary to extend the abstract syntax with the annotation attribute. Therefore, also a second option has been included: By using the separate "Semantic Annotation Model", the annotations can also be defined without modifying the underlying business process language and the annotation is stored separately from the conceptual models and ontology models. This uncoupling of conceptual models, annotations, and ontology models also provides a further advantage in terms of flexibility: By technically separating the annotations from the original models, the annotations do not affect the original process models and can be treated independently.

3.2 Semantic Annotations for Business Process Benchmarking

In the following we describe three particular aspects of using semantic annotations in business process benchmarking: For the *annotation* of business process and organizational models during data preparation, for the *analysis* of process data based on these annotations, and for the *semantic obfuscation* of data.

To show how semantic annotations can be used for benchmarking, we will use the example of determining the number of media disruptions in a particular existing process model. It is assumed that an existing process model shall be complemented with semantic annotations in order to integrate the additional knowledge about the occurrence of media disruptions. Therefore, we regard a segment of a business process that contains two instances of activities (see figure 3): The name attribute of the first activity instance is filled with the value "Check fax order from customer" and the name attribute of the second instance with "Enter order information in booking application". It is further assumed that

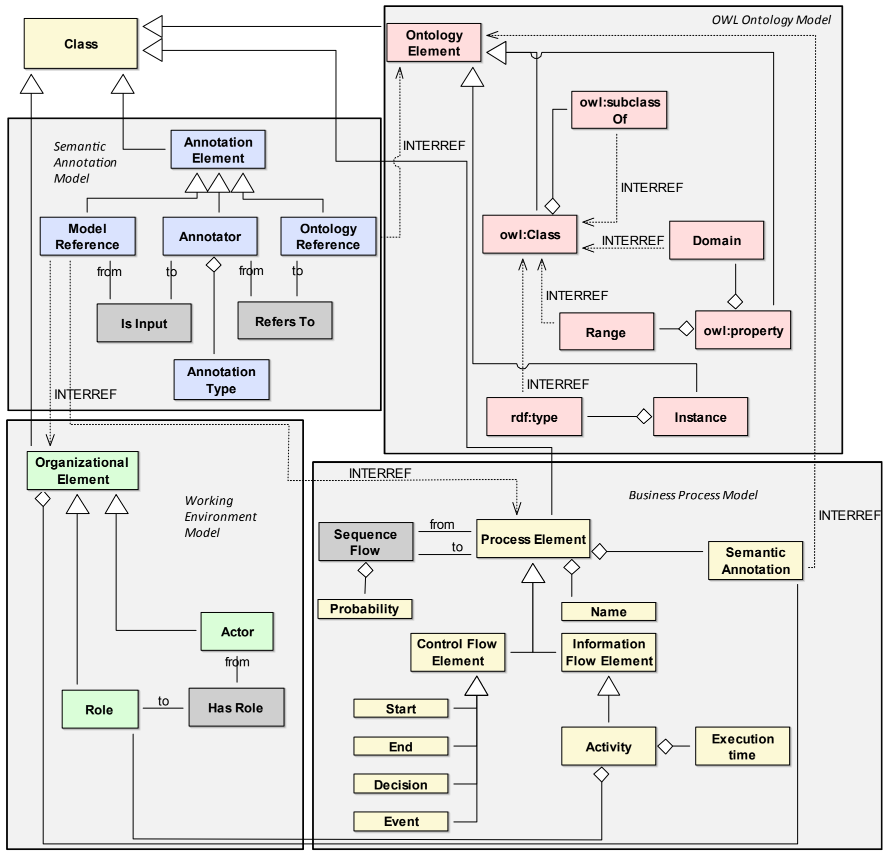
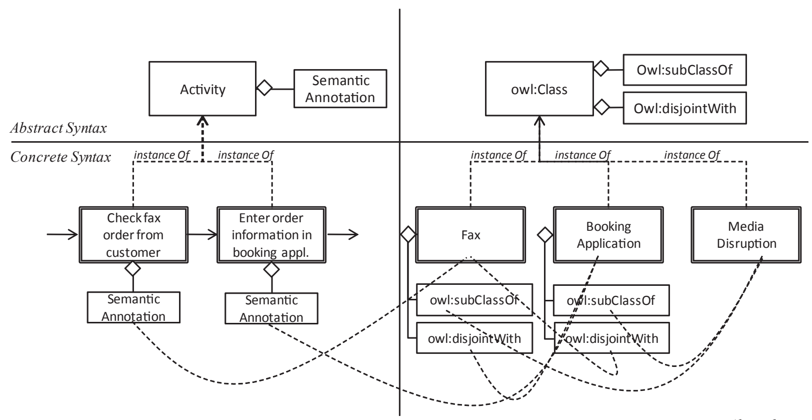


Fig. 2. Excerpt of the Meta Model for Semantic Annotations of Business Processes

the activities are connected by the "Sequence Flow" relation, which is shown in figure 3 by an arrow between the two instances.

At the same time, an ontology model is defined that contains three instances of "owl:Class". Thereby, ontology concepts are defined with the following URIs - for enhancing readability we leave out the preceding namespace definitions: "Media Disruption", "Booking Application", and "Fax". "Fax" and "Booking Application" are defined as sub classes of "Media Disruption" by adding the reference in the "owl:subClassOf" attribute. Due to the open world assumption used by OWL, the two sub classes of "Media Disruption" have to be explicitly defined as being disjoint. Therefore, the "owl:disjointWith" attributes of both sub classes are filled with a reference to the respective other sub class. The actual annotation of the activity instances can now be accomplished by adding the according references to the "Semantic Annotation" attributes.

With these definitions in place we can now outline how an algorithm can be implemented that computes the number of media disruptions in a process model (see figure 3). The depicted algorithm assumes that the process only consists of a sequence of activity instances that are stored in the array "x[]". It then iterates over this sequence of activities and determines whether there is a change in the semantic annotations that have to be sub classes of "Media Disruption" and disjoint from the next annotation. Until here, the use of OWL for describing the ontology has offered some advantages by providing useful axioms such as subClassOf and disjointWith. However, the formal semantics available for OWL provide additional options. Suppose one would like to add more specific types of booking applications. In this case the ontology could be easily extended by adding sub classes to "Booking Application". An automatic OWL reasoner can then check if the ontology is consistent with the previously defined concepts, e.g. that only disjoint concepts are used. At the same time, the outlined algorithm is still applicable based on another result by the reasoner: As sub class relationships in OWL are transitive, it can be inferred that also all sub classes of "Booking Applications" are sub classes of "Media Disruption".




```
SET i=0
SET x[]:instanceOf("Activity")
WHILE x[i].hasNext
    IF (x[i].SemanticAnnotation.subClassOf("MediaDisruption") &&
        x[i+1].SemanticAnnotation.subClassOf("MediaDisruption")) &&
        x[i].SemanticAnnotation.disjointWith(x[i+1].SemanticAnnotation)
    THEN
        SET MediaDisruption++;
    ENDIF
    i++;
ENDWHILE
```

Fig. 3. Illustration of Computations based on Annotated Models

The hierarchical character of OWL ontologies can also serve another requirement in benchmarking: As an ontology typically contains general concepts that are specialized into more specific ones, this information can be used to abstract information. This can be applied for meeting the requirements of confidentiality when sharing process data for benchmarking purposes. As shown in figure 4, the annotation of a process element (Step 1) allows the semantic abstraction to more general concepts (Step 2) which can then be assigned as a new name for the corresponding element (Step 3).

We denote this type of abstraction as *semantic obfuscation* because it does not completely remove the semantic information. It therefore still allows conducting semantic analyses of the underlying processes at least to a certain level. Furthermore, the reference to the original process models can still be preserved. In this way an external evaluator can analyze the obfuscated models and run algorithms on them while still being able to give feedback on where particular process parts may need to be improved. The example in figure 4 shows an excerpt of the account opening process at a bank. By annotating the process elements with concepts from an ontology (Step 1), the information can be abstracted based on the hierarchy defined in the ontology (Step 2), and the new information assigned to the corresponding model element (Step 3). Depending on the degree of obfuscation the user chooses, the corresponding higher level ontology concept is used as a replacement for the name attribute. Although the semantic information is then only available in an abstracted form, the remaining process information such as execution times or transition probabilities are fully preserved.

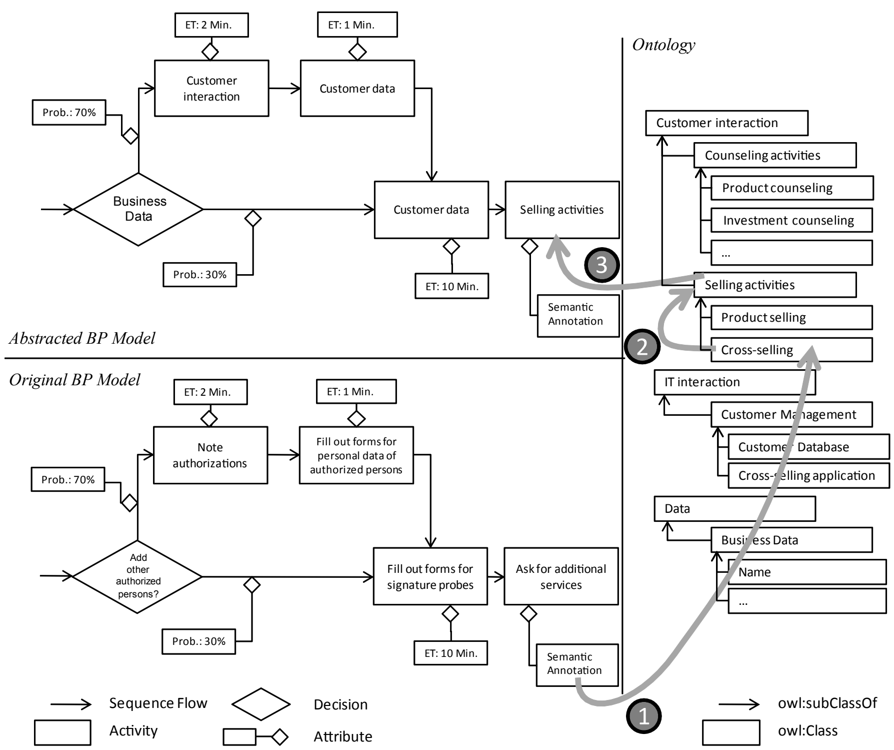


Fig. 4. Illustration of Semantic Obfuscation for Business Process Models

4 Implementation and Application

We have implemented a model editor for the described meta model by using the ADOxx meta modeling platform, which provides the scripting language ADOscript, the generic model query language AQL as well as import and export functionalities for exchanging models [26][1]. ADOxx was chosen primarily based on its industry-ready scalability and existing knowledge in regard to its implementation languages on the side of the authors. Furthermore, to support the semantic obfuscation of models we have implemented an algorithm in ADOscript that allows the automatic abstraction of the name attributes in the business process models by accessing the "owl:subClass" information provided in the OWL models. Thus, the algorithm is able to abstract each referenced concept by stepping up the class hierarchy and then assigning the name of the upper concept together with a unique ID as a label to the original element. The algorithm currently only supports single inheritance relationships, which proved however sufficient so far during the first evaluation.

In order to integrate OWL ontologies, a coupling between the Protégé ontology toolkit[2] and the ADOxx platform was established. The exchange of ontology information is thereby realized via a plugin for Protégé that translates the ontology information into the generic XML format of the ADOxx platform. In this way, the generic AQL query language can be used to retrieve information about the semantic annotations and the according ontology models. At this stage the resulting tool can be used for creating business process and working environment models and annotate them with the concepts from the imported OWL ontologies. By using the AQL query language, models can then be analyzed and compared.

For a first evaluation we applied the tool to two scenarios: The first one is based on the benchmarking of service interaction processes for the Bulgarian and Romanian chambers of commerce that have been previously elaborated in the LD-CAST project[3] (see figure 5). To enable the semantic annotation of these processes, a specific benchmarking ontology was first developed in Protégé. It provides a simplified description of the domain of business process benchmarking and contains in particular the OWL classes "Automated_business_process_task", "Business_process_task_with_media_disruption", and "Manual_business_process_task". The ontology was then translated into the ADOxx XML format and made available as an OWL model. The annotation of the service processes could be easily accomplished by linking the activity elements to the corresponding ontology concepts. By using the AQL query language, the statistics for the benchmarking in regard to the degree of automation and the occurrence of media disruptions could then be successfully retrieved. As an example the syntax of an

[1] The implementation will be made freely available for further evaluation in the course of the SeMFIS project of the Open Model Initiative at http://www.openmodels.at/web/semfis

[2] See http://protege.stanford.edu

[3] LD-CAST stands for Local Development Cooperation Actions Enabled By Semantic Technology, http://www.ldcastproject.com

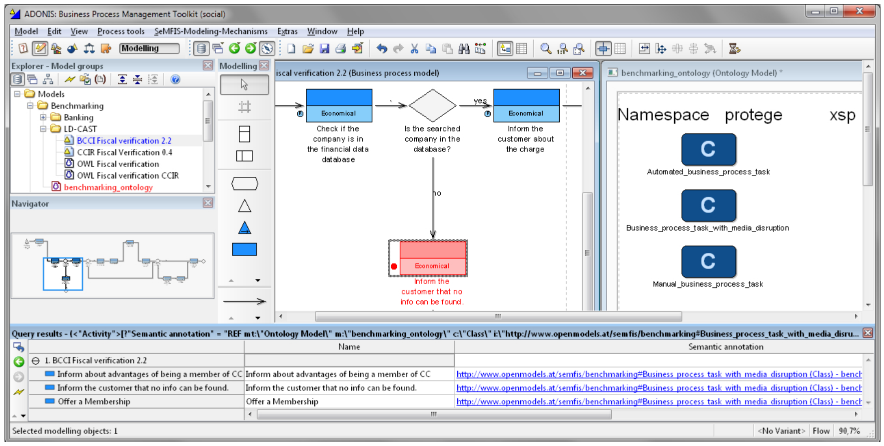


Fig. 5. Screenshot of the ADOxx Implementation

AQL query for retrieving Activity elements from a Business Process Model that have been annotated with the OWL Class "Business_process_task_with_media_disruption" is given in the following:

Sample AQL Query:

```
(<"Activity">
[?"Semantic annotation" =
  "REF mt:\"Ontology Model\"
   m:\"benchmarking_ontology\"
   c:\"Class\"
   i:\"http://www.openmodels.at/semfis/benchmarking#
   Business_process_task_with_media_disruption\""])
```

The second scenario comprised the application of the semantic obfuscation approach to the banking domain. Based on two processes for the opening of accounts at two Swiss banks that are publicly available and have been further refined by the author based on their documentation[4], the obfuscation algorithm could be applied. For this purpose, a simplified domain ontology for banking was elaborated in Protégé. This ontology comprised in particular several subclass and superclass relationships, e.g. by using a general concept such as "Selling_activites" and a specialized concept "Cross-selling_activity". In comparison to the first scenario, the semantic annotation in this case required more effort as all necessary domain concepts had to be assigned to the activities to accurately describe the content. After the annotation the obfuscation algorithm could be successfully applied which led to the outcomes shown in figure 6.

[4] The processes have been elaborated and published by Gerardo Palmisano for the Hypothekarbank Lenzburg and by Jonas Winkler for the Spar- und Leihkasse Riggisberg on http://www.lernender.ch

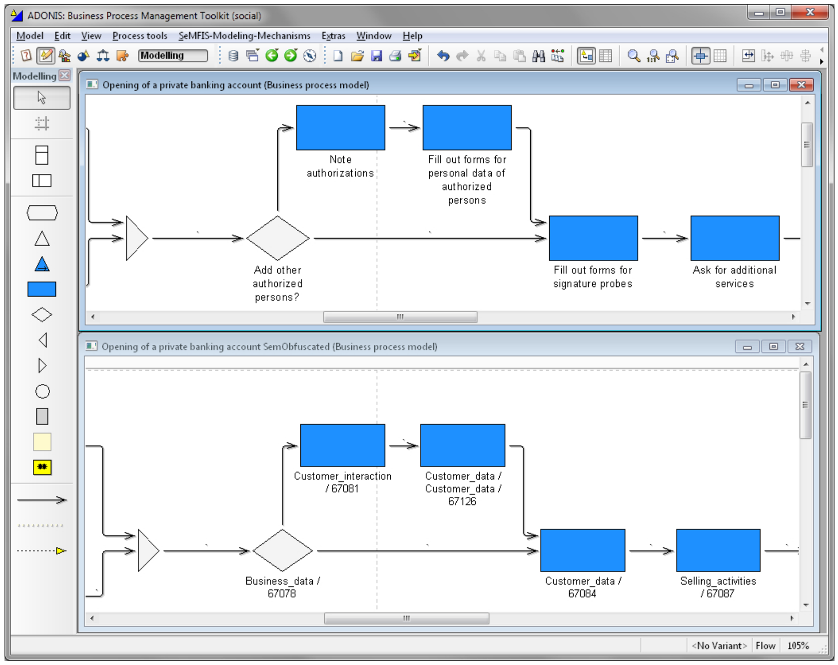


Fig. 6. Example for the Semantic Obfuscation Using an Account Opening Process

5 Related Work

When comparing our approach to existing work, three categories of related approaches can be identified: Approaches that deal with IT-based methods for *business process benchmarking*, approaches that describe *the application of semantic annotations of process models and according tools*, and *semantic schemata for process evaluation*. For the first category, the approach by [9] who describe a logic based approach for the comparison of business processes to support benchmarking, is closely related to our approach. However, in contrast to our approach a common data dictionary is assumed to already exist which does not provide the expressiveness, and shareability of OWL ontologies. Similarly, the approaches by [16] who describe a method to construct comparable business process models using a domain specific language and [19] who present a method to create naming conventions for arbitrary conceptual models share several aspects. As already pointed out in section 2.3, our approach does however not enforce a particular modeling language but instead uses additional, decoupled semantic information that is machine-processable.

In regard to approaches that discuss the use of semantically annotated business process models several related approaches exist. These include but are not limited to the detection of regulatory compliance [27, 28], semantic reverse business engineering in order to analyze productive ERP systems [12, 29], semantic model comparisons [23], to support cross-organizational business processes and interoperability [30] or for the dynamic binding of web services during the execution of business processes [21]. These approaches may be directly used in addition to our approach, e.g. to add performance indicators that measure the compliance of processes, to map the data acquired during the execution of processes into the process models, to identify similar processes prior to benchmarking, to integrate different types of modeling languages or to automatically plan and execute processes based on the insights gained through benchmarking.

Concerning the tool support for semantic annotations of process models several options are available: These include WSMO studio [31] that has been specifically developed in the SUPER-IP project in regard to automated process execution, the semantic extensions for the Maestro BPMN tool [22] and ARIS [21], and SemPeT for realizing semantic annotations for Petri nets [23]. Although these tools may also be used for applying our approach, they were either not available due to their licenses or used specific modeling and ontology languages that are different to the ones we proposed in our approach. However, apart from licensing issues it should be possible to adapt these tools to support our approach as well.

Semantic schemata that can be used for the annotations include not only approaches based on formal semantics such as the ontology proposed for key performance indicators [32], the Core Ontology for Business Process Analysis (COBRA) [33], the OWL based business process management ontology (BPMO) by [34] or the business process ontology (BPO) by [35], but also semi-formal approaches such as the schema for monitoring and analyzing processes by [36]. Furthermore, any kind of domain ontology that contains a hierarchical structure necessary for the semantic obfuscation may be applied. In order to directly use them for our approach they would have to be either available in the OWL format or be translated to it.

6 Conclusion and Outlook

With the proposed approach of using semantic annotations for business process models it could be shown how these additional technical functionalities can help to prepare process data for benchmarking and conduct machine-based semantic process analyses. The approach can be easily extended to other model types, e.g. by accessing the information in the organizational model and obfuscating the names and roles of particular persons that participate in a business process or by additional resource models, e.g. for analyzing the IT usage in a business process. Apart from the functionality presented here, the next step will be to apply the approach in practice and conduct according empirical research and user studies to further evaluate its applicability. Thereby the techniques for accomplishing the annotations shall be further detailed based on the feedback from domain experts. Furthermore, the distribution of model content using the discussed semantic obfuscation technique will have to be further evaluated in terms of maintaining confidentiality for conducting benchmarks in practice.

7 Acknowledgement

The work on this paper has been funded by the Austrian Science Fund (FWF) in the course of an Erwin-Schrödinger-Fellowship grant number J3028-N23.

References

1. Camp, R.: Business Process Benchmarking: Finding and Implementing Best Practices. ASQC Quality Press, Milwaukee, Wisconsin (1995)
2. Andersen, B.: Benchmarking. In Rolstadas, A., ed.: Performance Management - A business process benchmarking approach. Chapman and Hall, London et al. (1995) 211–242
3. Amaral, P., Sousa, R.: Barriers to internal benchmarking initiatives: An empirical investigation. Benchmarking: An International Journal **16**(4) (2009) 523–542
4. Karagiannis, D., Woitsch, R.: The PROMOTE approach: Modelling Knowledge Management Processes to describe an organisational KMS. In: ECAI Workshop, Lyon, France, INRIA (2002)
5. Hill, J.: Highlights From the Business Process Improvement Track at Symposium/ITxpo, Gartner Research - www.gartner.com (2010)
6. Legner, C., Oesterle, H.: Process benchmarking - A methodological approach for process development using standard software (in German). In: 4th International Conference on Business Informatics. Physica, Heidelberg, Germany (1999)
7. List, B., Korherr, B.: An Evaluation of Conceptual Business Process Modelling Languages. In: SAC'06, Dijon, France, ACM (2006)
8. Hinton, M., Francis, G., Holloway, J.: Best practice benchmarking in the UK. Benchmarking: An International Journal **7**(1) (2000) 52–58
9. Juan, Y., Ou-Yang, C.: A Process Logic Comparison Approach to Support Business Process Benchmarking. Int. Journal of Advanced Manufacturing Technologies **26** (2005) 191–210
10. Heinrich, R., Paech, B.: Defining the Quality of Business Processes. In: Modellierung 2010. GI LNI, Klagenfurt (2010) 133–148
11. Karagiannis, D., Kuehn, H.: Metamodeling Platforms. In: Third International Conference EC-Web 2002 Dexa 2002. Springer (2002) 182
12. Celino, I., De Medeiros, A., Zeissler, G., Oppitz, M., Facca, F., Zoeller, S.: Semantic business processes analysis. In Hepp, M., Hinkelmann, K., Karagiannis, D., Klein, R., Stojanovic, N., eds.: SBPM 2007 Workshop. Volume 251., CEUR Workshop Proceedings (2007)
13. Mylopoulos, J.: Conceptual Modeling and Telos. In Loucopoulos, P., Zicari, R., eds.: Conceptual Modelling, Databases and CASE: An Integrated View of Information Systems Development. Wiley (1992) 49–68
14. Rosemann, M.: Potential pitfalls of process modeling: Part A. BPM Journal **12**(2) (2006) 249–254
15. Cardoso, J., Mendling, J., Neumann, G., Reijers, H.: A discourse on complexity of process models. In: BPM Workshops. Springer (2006) 115–126
16. Becker, J., Breuker, D., Pfeiffer, D., Raeckers, M.: Constructing comparable business process models with domain specific languages - an empirical evaluation. In: ECIS 2009, Verona, Italy (2009)
17. Peleg, M., Tu, S.: Design Patterns for Clinical Guidelines. AI in Medicine **47**(1) (2009) 1–24

18. Schuette, R., Rotthowe, T.: The Guidelines of Modeling - An Approach to Enhance the Quality in Information Models. In: ER'98. Springer (1998) 240–254
19. Becker, J., Delfmann, P., Herwig, S., Lis, L., Stein, A.: Towards Increased Comparability of Conceptual Models - Enforcing Naming Conventions through Domain Thesauri and Linguistic Grammars. In: ECIS 2009, Verona, Italy (2009)
20. Thomas, O., Fellmann, M.: Semantic Business Process Management: Ontology-based Process Modeling Using Event-Driven Process Chains. IBIS **2**(1) (2007) 29–44
21. Stein, S., Stamber, C., El Kharbili, M.: ARIS for Semantic Business Process Management. In: BPM Workshops. Volume 17 of LNBIP., Springer (2009) 498–509
22. Born, M., Hoffmann, J., Kaczmarek, T., Kowalkiewicz, M., Markovic, I., Scicluna, J., Weber, I., Zhou, X.: Semantic Annotation and Composition of Business Processes with Maestro. In: ESWC, Springer (2008)
23. Ehrig, M., Koschmider, A., Oberweis, A.: Measuring Similarity between Semantic Business Process Models. In Roddick, J., Hinze, A., eds.: APCCM 2007. Volume 67. ACM (2007) 71–80
24. W3C: Owl web ontology language - overview w3c recommendation 10 february 2004 (2004) `http://www.w3.org/TR/owl-features/` accessed 16-09-2005.
25. Horrocks, I., Patel-Schneider, P., Van Harmelen, F.: From SHIQ and RDF to OWL: The Making of a Web Ontology Language. Web Semantics: Science, Services and Agents on the World Wide Web **1**(1) (2003) 7–26
26. Fill, H.G.: UML Statechart Diagrams on the ADONIS Metamodeling Platform. Electronic Notes in Theoretical Computer Science **127**(1) (2004) 27–36
27. Governatori, G., Hoffmann, J., Sadiq, S., Weber, I.: Detecting Regulatory Compliance for Business Process Models through Semantic Annotations. In: 4th Business Process Design Workshop, Milan, Springer (2008)
28. El Kharbili, M., Stein, S., Markovic, I., Pulvermueller, E.: Towards a Framework for Semantic Business Process Compliance Management. In: GRCIS08 Workshop. (2008)
29. Oppitz, M., Zeissler, G., Zoeller, S., Carenini, A.: Semantic Reverse Business Engineering - Public Deliverable 5.7 of Project IST 026850 SUPER (2009) `http://www.ip-super.org/res/Deliverables/M36/D5.7.pdf` accessed 15-03-2011.
30. Hoefferer, P.: Achieving Business Process Model Interoperability Using Metamodels and Ontologies. In Oesterle, H., Schelp, J., Winter, R., eds.: ECIS2007, St. Gallen, Switzerland, University of St. Gallen (2007) 1620–1631
31. Dimitrov, M., Simov, A., Momtchev, V., Konstantinov, M.: WSMO Studio a Semantic Web Services Modelling Environment for WSMO (System Description). In: ESWC, Springer (2007)
32. Wetzstein, B., Karastoyanova, D., Leymann, F.: Towards Management of SLA-Aware Business Processes Based on Key Performance Indicators. In: 9th Workshop on Business Process Modeling, Development, and Support. (2008)
33. Pedrinaci, C., Domingue, J., De Medeiros, A.: A core ontology for business process analysis. In: ESWC, Springer (2008)
34. Jenz & Partner: The Open Source Business Management Ontology (BMO) (2006) `http://www.bpiresearch.com/Resources/RE_OSSOnt/re_ossont.htm` accessed 13-06-2006.
35. Markovic, I., Costa Pereira, A.: Towards a formal framework for reuse in business process modeling. In: BPM Workshops, Springer (2007) 484–495
36. Karagiannis, D., Ronaghi, F., Fill, H.G.: Business-oriented IT management: developing e-business applications with E-BPMS. In: ICEC 2008. Volume Vol. 258., ACM (2007) 97–100