Semantic Audit Application

Katalin Ternai, Ildikó Szabó

Department of Information Systems, Corvinus University of Budapest
1093 Budapest, Fővám tér 13-15., HUNGARY

katalin.ternai@uni-corvinus.hu
iszabo@informatika.uni-corvinus.hu

- **Abstract.** Compliance checking of business processes executed by auditors requires to analyze documents e.g. log files, business process models depending on requirements derived from reference guidelines. This paper presents a forward compliance checking application for facilitating conformant behavior by detecting organizational operations and their deviations based on these documents in a semantic way. This application has been tested on the Internalization process in the respect of Erasmus mobility.

Keywords: erasmus mobility, process ontology, ontology learning, ontology matching, compliance check

1 Introduction

Auditors have to collect evidence whether organizational operations are working according guidelines and well-documented. Computer-aided audit tools are not capable of processing documents in a semantic way [1]. Different semantic approaches exist for solving this compliance checking. Supporting compliant process executions or detecting non-compliant process executions are distinguished as forward or backward compliance checking [2]. This is only one dimension of the semantic framework elaborated in the SUPER project for compliance management. This public research project presents five perspectives on compliance checking: design-time/run-time; forward/backward; active/passive; task checking/process checking or engine-based/query-based perspective [3]

Auditors must detect current organizational operations and their deviations through scrutinizing documents. Our solution wants to help auditors in this work. We assume that business process models are hidden within documents. Process ontologies preserve the structure of these models and easily filtered in a semantic way. Process ontologies are a set of abstract entities like *Process step (S), Data (D), IT System(T)* and *Roles(R)* classes. The function M(X) is to determine the set of instances of X class. The linkage between instances of these classes can be considered as a graph P = (N, L), where N= $M(S) \cup M(D) \cup M(T) \cup M(R)$ and L contains the following ordered pairs. This table contains their interpretations.

Ordered pairs	Interpretation of them
performed_by (M(S), M(R))	A given role is responsible for executing a given process step

Produces_output(M(S),M(D))	A given data (including a document) is created by a process step
Uses_input(M(S),M(D))	A given process step uses an input for its run
Uses_system(M(S), M(T))	A given process step uses an information system for its run

Table 1. Relations of our process ontologies

In our solution, business process models in the shape of process ontologies are extracted from organizational documents with using standard process models in the form of process ontologies as well. The structure and objects of standard process models are used as compliance specifications because, in our approach, these processes must be run if conformant behavior is followed by a given organization. The deviations can be discovered by investigating the structure and objects of the above-mentioned process ontologies. Ontology matching is capable of performing this kind of structural and semantic examination. Hence an ontology matching tool is used to detect deviations of organizational processes from standard ones and create a transparent report from them for auditors. This forward compliance checking is executed *in design time* without using any runtime data, *on process level* instead of task level and with embodied ontology-based techniques. Ontology-based approach was used by [4], but their compliance ontology was used to determine rules and not for executing the matching. Our Semantic Audit Application is presented in the next section.

2 Semantic Audit Application

The process ontology building component is responsible for discovering semantic contents in documents in an automated manner. This module uses reference process ontologies transformed from BOC ADONIS[1] process models by using XSLT transformation as inputs. These process ontologies and business process models are stored in the repository of this system. The first step of the algorithm is to build the organizational process ontology (OPO) with the help of the reference process ontology (RPO). The skeleton of these ontologies are the same. *Semantic rules* are used to find process elements (like IT resources, roles, input/output documents). First of all, the name of these process elements (like Student, Coordinator as Roles) are tried to find within the document. If it does, the initial OPO will be extended by these entities as classes. New process elements will be discovered by using the initial relations of the process ontologies (see in Table 1) as open queries. For example the goal is to find a role who performs a process step. The meaning of „performed_by“ relation suggests that “<x>by the<y>” open query as pattern within text can detect a relation between an x task (process step) and an y role. Newly discovered process elements will be added to the OPO.

[1] https://uk.boc-group.com/adonis/

The algorithm picks each process steps from the RPO in order to add process steps to the OPO. It splits the name of these process steps into terms. It seeks them through texts and measures the frequency of their occurrence within a given sentence. The sentence providing the higher value will validate this process step. Having identified these process steps, they are added to the OPO and connected to an existent other process elements of the OPO (Role, IT resources etc) which are located nearby (namely within a given radius). The process ontology building component creates the organizational process ontology containing process elements from the organizational document.

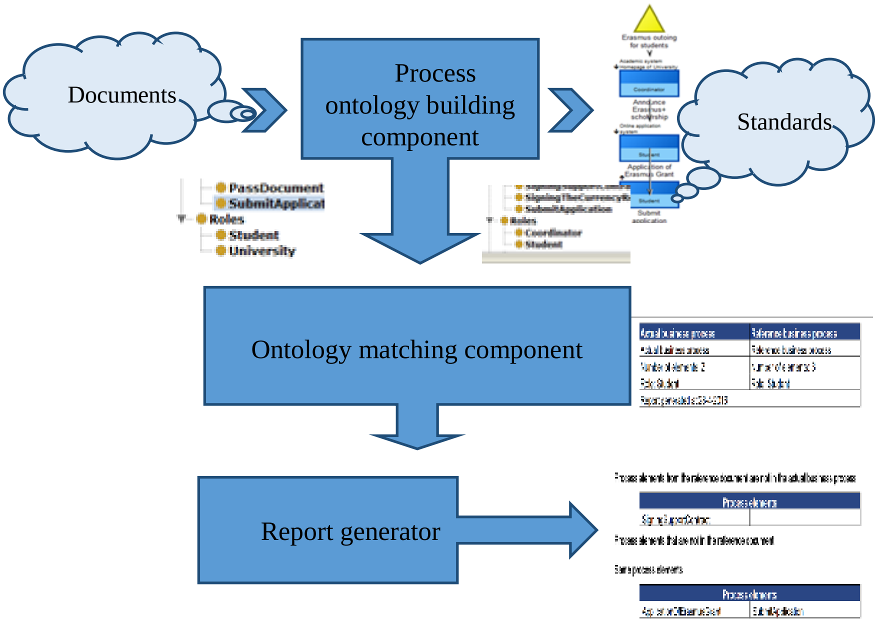

Fig. 1. Components of the Semantic Audit Application

The ontology matching component uses DL Queries and Protégé 4 OWL Diff for filtering the ontologies, adding instances with a predefined attribute (e.g. region) to them and creating a technical report after executing the matching. This report is processed by **a report generator** to create a transparent report for auditors which contains information about the number of task, filtered role, missing, unnecessary or common organizational process elements. Hence auditors can discover areas requiring deeper investigations in the next phase when leaders are interrogated by them.

3 Conclusion and future work

Nowadays Campus Mundi projects are to improve higher education processes in Hungary. The audit guideline elaborated for investigating compliance checking of Internationalization activities wants to detect “how the current mechanisms are effective”. Our semantic audit application can help to compare institutional processes with standard processes articulated in the Erasmus Mobility Handbook. The Student Application procedure was used to test this application. Erasmus mobility calls represent the organizational documents. This test was executed on ten different sources.[2] The precision of this application is presented by the following charts. The first one shows that how many total and meaningful roles were detected by the above-mentioned algorithm using “by the” semantic rule. It seems that at most one wrong role was extracted in the most cases. The blue line on the second chart presents how many tasks related to the Student role were extracted from the organizational documents, the red line shows how many of them were the same as in the reference process. The notable differentiation between them indicates two cases: the algorithm identified tasks badly or the higher education institutions obligate students to perform tasks which are related to a different role in the handbook. The latter case must be investigated by the auditors, because the segregation of duties control might be failed. So the improvement of the algorithm leads to emerge an application that provides auditors reports for facilitating further investigations. This JAVA application containing libraries of OWL API, DLQueryExample and the SVN Repository of Protégé 4 OWL Diff.

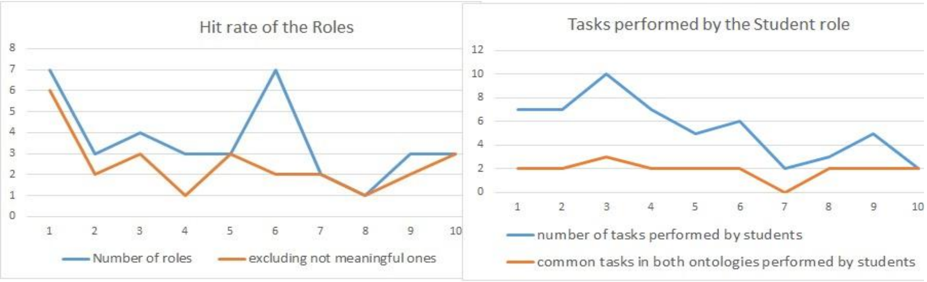

Fig. 2. Results of the application

4 Reference

1. Braun RL, Davis HE (2003) Computer-assisted audit tools and techniques: analysis and perspectives. Managerial Auditing Journal 18:725–731. doi: 10.1108/02686900310500488
2. Ramezani E, Fahland D, Aalst WMP van der (2012) Where Did I Misbehave? Diagnostic Information in Compliance Checking. In: Barros A, Gal A, Kindler E (eds) Business Process Management. Springer Berlin Heidelberg, pp 262–278
3. El Kharbili M, Stein S, Markovic I, Pulvermüller E (2008) Towards a framework for semantic business process compliance management. In: Proceedings of the workshop on Governance, Risk and Compliance for Information Systems. pp 1–15
4. Schmidt R, Bartsch C, Oberhauser R (2007) Ontology-based Representation of Compliance Requirements for Service Processes. In: SBPM.

[2] http://erasmus.yasar.edu.tr/student-mobility/outgoing-students/
http://www.ciim.ac.cy/outgoing-students
https://www.international.tum.de/en/going-abroad/students/erasmus/...etc.