



An Efficient Business Process Compliance Checking Approach

Jörg Becker, Philipp Bergener, Dominic Breuker, Patrick Delfmann, Mathias Eggert

► To cite this version:

Jörg Becker, Philipp Bergener, Dominic Breuker, Patrick Delfmann, Mathias Eggert. An Efficient Business Process Compliance Checking Approach. Governance and Sustainability in Information Systems: Managing the Transfer and Diffusion of IT (Working conference), Sep 2011, Hamburg, Germany. pp.282-287, 10.1007/978-3-642-24148-2_19. hal-01571720

HAL Id: hal-01571720

<https://inria.hal.science/hal-01571720>

Submitted on 3 Aug 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

An Efficient Business Process Compliance Checking Approach

Jörg Becker
Philipp Bergener
Dominic Breuker,
Patrick Delfmann,
Mathias Eggert

University of Muenster, ERCIS – European Research Center for Information Systems,
Leonardo-Campus 3, 48149 Münster, Germany
+49 (0) 251 8338 100
{becker|bergener|breuker|delfmann|eggert}@ercis.uni-muenster.de

Abstract. Assuring compliant business processes is an important task of business process management, which is commonly supported by the use of business process models. As every compliance rule corresponds with a typical structure, the detection of those corresponds to a pattern matching problem. More specifically, we encounter the problem of subgraph isomorphism. In this paper we propose an automatic business process compliance checking approach that relies on a subgraph isomorphism algorithm and that is suitable for process models in general. As common subgraph isomorphism is a problem that can only be solved in exponential time, we use an algorithm that simplifies the problem through pre-processing. This makes the isomorphism solvable in polynomial time. With the approach, we aim at supporting decision makers in business process compliance management.

Keywords: Compliance, Business Process Management, Pattern Matching, Subgraph Isomorphism, Efficiency.

1 INTRODUCTION

The financial crisis has demonstrated impressively how difficult it is to adhere to legal regulations and internal as well as external compliance requirements concerning business processes. The financial crisis even aggravates the complexity by introducing more and tighter regulations for financial institutions (Caldwell 2009; Caldwell, Bace and Lotto 2009; Opromolla 2009). This trend towards more regulation has also had an impact on IS research. Throughout the last years, an increasing number of approaches to solve compliance issues were published in this research area (Abdullah, Indulska and Sadiq 2009).

In this paper, we introduce a flexible model-driven business process compliance checking approach. It allows for specifying (nearly) any type of compliance rule,

whose instances can be searched in any type of conceptual model. For this purpose, we make use of an efficient subgraph isomorphism (Ullmann 1976) approach. Its general applicability distinguishes our approach from existing ones. As the general subgraph isomorphism problem is only solvable in exponential processing time (Garey and Johnson 1979), according algorithms are inefficient and not suitable for large process models. Therefore, we propose reusing a subgraph isomorphism algorithm that makes use of pre-processing, making the isomorphism itself solvable in polynomial time. A precondition for consistent pattern matching results is that the process models are semantically unambiguous. Therefore, we restrict our approach to business process models that are semantically standardized.

2 RELATED WORK

2.1 Business Process Compliance Checking

The concept of business process compliance denotes that the execution of certain processes complies with a set of regulations (Sadiq, Governatori and Namiri 2007). Kharbili et al. (2008) classify the implementation of control mechanisms in three time-dependent phases “Design-Time Compliance Checking”, “Runtime Compliance Checking” and “Backward Compliance Checking”. In this paper, we focus on Design-Time Compliance Checking.

It is related to modeling compliance rules and process models. Notable research in this field has been done by Governatori et al. (2008), Governatori et al. (2010), Stijin et al. (2006), Wörzberger et al. (2008), Sadiq et al. (2007), Liu et al. (2007). The approaches in this phase mainly focus on the extension or creation of modeling languages that are able to represent compliance rules or might be used for (semi-)automated compliance checking. One example is the Formal Contract Language (FCL) (Sadiq et al. 2007). Within this approach, the identified controls are transformed into control rules and classified with particular control tags. These control tags combine control rules with process models. This allows for (automatic) assignment to corresponding process elements (Sadiq et al. 2007).

Our approach differs from existing ones inasmuch it is generic and thus it abstracts from specific modeling languages. It is applicable to any modeling language that can be formalized as a graph (Harary 1969). This means that both the models and the compliance rules are regarded as graphs, where the compliance rule graph is searched in the model graph. Searching for (small) graphs in (large) graphs corresponds with the problem of subgraph isomorphism known from algorithmic graph theory (Ullmann 1976). Regarding compliance checking as a subgraph isomorphism problem allows for representing compliance rules of arbitrary structure. This aspect constitutes another difference to existing approaches.

2.2 Subgraph Isomorphism

One of the most widely known matching algorithms is Nauty, developed by McKay (1981). This algorithm is based on group theory and computes the automorphism group of both the search graph and the subgraph which is then used to create a canonical labeling defining a node ordering. Subgraph isomorphism can be found by finding identical adjacency matrices of the two graphs in canonical form. This last verification step can be computed in $O(n^2)$ time. The construction of the canonical labeling, however, is a preprocessing step requiring exponential time in the worst case.

Messmer et al. (2000) extend this approach by proposing an algorithm for finding isomorphisms in a library of graphs. Their approach is based on a recursive decomposition of each graph into smaller subgraphs until the decomposed graphs consist of single nodes. The matching algorithm then exploits the fact that some decomposed graphs are identical and avoids redundant comparisons. In another work, the same authors include a decision tree into their algorithm that allows for matching an input graph against an entire library of graphs in $O(n^2)$ time with n being the size of the input graph (Messmer and Bunke 1999). This algorithm exploits both the facts that the underlying graphs are labeled and directed and accomplishes the subgraph search with impressive speed. While the construction of a decision tree is rather costly, the effort can be justified assuming that the processes in which patterns shall be searched do not change very often. Given this assumption, the decision tree has to be recomputed infrequently. This is why we chose this algorithm as a basis for our compliance checking approach (cf. Conte et al. (2004) for a detailed description of related algorithms).

3 PATTERN MATCHING

3.1 Definition of Compliance Rules

In general, there are an infinite number of compliance rules depending on the actual domain and subjective perception. As such, we are not able to explain and provide an exhaustive list of identification heuristics for all of the rules. Instead, we show an exemplary excerpt which is based on the German money laundry law (ger. Geldwäschegesetz, GWG) (cf. Becker et al. (2010) for a comprehensive list of compliance rules).

Two sample compliance rules are depicted as subgraphs in Figure 1. A sequence requires a specific activity (here: “B”) to be placed between two other subsequent activities (here: “A” and “C”). The corresponding subgraph is defined as $GS=(VS,ES)$ with $VS=\{A,B,C\}$ and $ES=\{(A,B),(B,C)\}$. Separation of duties requires two subsequent activities (here: “A” and “B”) to be performed by two different people (here: “P1” and “P2”). The corresponding subgraph is defined as $GS=(VS,ES)$ with $VS=\{A,B,P1,P2\}$ and $ES=\{(A,B),(P1,A),(P2,A)\}$. Different types or names of the subgraph’s vertices are realized by using a labeled graph. Unique names stored in the graph’s labels are realized through semantic standardization (cf. previous section).

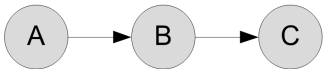
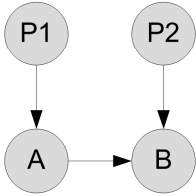
Compliance Pattern	Sequence	Separation of Duties
Corresponding Subgraph	 <pre> graph LR A((A)) --> B((B)) B --> C((C)) </pre>	 <pre> graph TD P1((P1)) --> A((A)) P2((P2)) --> B((B)) A --> B </pre>

Fig. 1. Examples of Compliance Rules and Corresponding Subgraphs

3.2 The Messmer-Bunke-Algorithm for sub-graph isomorphism

To detect such compliance rule patterns in a library of process models we propose using an adapted version of the algorithm for subgraph isomorphism presented by MESSMER AND BUNKE in (Messmer et al. 1999). The algorithm operates on a number of a priori known model graphs (in our case the process models) and an input graph (the pattern). The basic idea of the algorithm is to compute all possible permutations of the adjacency matrices of the model graphs, with the vertex labels on the main diagonals of the adjacency matrices, and to arrange them in a decision tree during a pre-processing step. At runtime the adjacency matrix of the input graph is used to find identical matrices in the decision tree. The corresponding permutation matrices represent the searched subgraph isomorphism. The exact algorithm is given in (Messmer et al. 1999).

3.3 Algorithm Adaptation

There are a few modifications necessary to account for particularities of our patterns. First, due to the possibility to incorporate wildcard elements standing for a number of arbitrary activities and prohibited activities into a pattern, we do not solve the traditional subgraph isomorphism problem but rather a modified one in which two nodes of a pattern may be connected by a path of length k instead of being directly connected via a single edge.

To cope with this challenge we compute, for each of the model graphs, a distance matrix containing a vector of distances of all paths with maximum length k between any combinations of nodes. For each of these paths, we also associate a list of labels encountered on that path. During the creation of the decision tree, the distance matrix is treated analogously to the adjacency matrix. Thereby, for each row column element of the adjacency matrix that is associated to a node of the search graph, we also associate an appropriately permuted row column element of the distance matrix. If we now encounter a wildcard while searching the tree for a subgraph occurrence, we look up the entry of the distance matrix instead of the adjacency matrix and check if

there exists a path being short enough. If the wildcard prohibits the occurrence of certain labels on the path, we additionally check if the candidate path is free of such labels.

Another problem of the original algorithm is that it solves the induced subgraph problem. That is, it requires for each edge between nodes in the model graph the presence of an edge between the associated nodes in the pattern. However, we are interested in any match of our pattern regardless of additional edges between nodes of the model graph. To adapt the algorithm in such a way that it finds every pattern, we again need to extend the decision tree. The original tree has, on each level k , a node for each permutation of an adjacency matrix of all the subgraphs of size k . Our modification is that we do want to include not only all these subgraphs but, for each of the subgraphs GS , additional subgraphs being equal to GS except that an arbitrary set of edges has been removed. When searching a pattern we can then not only find an occurrence with nodes being connected exactly as it is specified in the pattern graph but we also find any occurrence having additional edges in the model graph by matching it to one of the newly incorporated tree nodes for which exactly these edges have been removed.

A third issue with the algorithm is that we interpret the building block type assigned to a vertex as its label. However, it might be that there are certain attributes of vertices that might be of relevance when defining a pattern. Consequently, a pattern should only be returned if, besides the labels of the vertices, the corresponding attributes of the vertices also match each other. A straightforward solution to this problem would be to interpret the combination of building block type and attribute as the label of a vertex. Unfortunately, this demands that, when creating a compliance rule pattern, a value has to be assigned to all available attributes of vertices. Thus, this solution is not an option, since many patterns will leave certain attribute values unspecified. The only other option left is to apply the algorithm with labels being building block types and to subsequently check each identified occurrence of the pattern for correctness of attributes. During this post-processing step, each identified subgraph whose attributes do not match the pattern is discarded.

4 OUTLOOK

In the short term, research will continue along two consecutive lines. We first intend to implement the algorithm in order to test it on a large library of business processes. In a second step we aim at defining a large set of compliance rule patterns. With these two libraries of both input graphs as well as process models we intend to conduct an empirical analysis of the algorithm's practical runtime behavior, since the algorithm has to prove its efficiency not only theoretically but also in practice. Results will point to further optimization potential of the extended algorithm.

Although the relevance of our approach was confirmed in preliminary discussions with compliance experts in the financial sector, the support of compliance management experts through the introduced approach must be shown in detail through qualitative and quantitative studies. First evidence from accompanying

research to this paper in conjunction with compliance experts suggests that the complexity of compliance management will be better handled and thus the effort for compliance checking will be lower by using the presented approach.

REFERENCES

- Abdullah, S.N., Indulska, M., and Sadiq, S. "A Study of Compliance Management in Information Systems Research," ECIS, Scholar One, Verona, 2009.
- Becker, J., Ahrendt, C., Coners, A., Weiss, B., and Winkelmann, A. "Business Rule Based Extension of a Semantic Process Modeling Language for Managing Business Process Compliance in the Financial Sector," *Lecture Notes in Informatics: Modellierung betrieblicher Informationssysteme (MobIs)* (175:1) 2010, pp 201-206.
- Caldwell, F. The Worldwide Economic Crisis will Bring Real-Time Reporting for Risk Management Gartner Research, Gartner, Inc., 2009.
- Caldwell, F., Bace, J., and Lotto, R.J.D. U.S. Financial System Regulatory Overhaul Brings More Scrutiny Gartner Research. Gartner, Inc., 2009.
- Conte, D., Foggia, P., Sansone, C., and Vento, M. "Thirty Years of Graph Matching in Pattern Recognition," *International Journal of Pattern Recognition and Artificial Intelligence* (18:3) 2004, pp 265-298.
- Garey, M.R., and Johnson, D.S. *Computers and Intractability: A Guide to the Theory of NP-Completeness* W. H. Freeman, 1979.
- Governatori, G., Hoffmann, J., Sadiq, S., and Weber, I. "Detecting regulatory compliance for business process models through semantic annotations," *Workshop on Business Process Design*, Milan, 2008.
- Governatori, G., and Rotolo, A. "A Conceptually Rich Model of Business Process Compliance," *APCCM*, Brisbane, 2010.
- Harary, F. *Graph Theory* Reading, 1969.
- Kharbili, M.E., de Medeiros, A., Stein, S., and van Der Aalst, W.M.P. "Business Process Compliance Checking: Current State and Future Challenges," *Lecture Notes in Informatics: Modellierung betrieblicher Informationssysteme (MobIs)* (141) 2008, pp 107-113.
- Liu, X., Müller, S., and Xu, K. "A static compliance-checking framework for business process models," *IBM Systems Journal* (46:2) 2007, pp 335-361.
- McKay, B.D. "Practical Graph Isomorphism," *Congressus Numerantium*:30) 1981, pp 45-87.
- Messmer, B.T., and Bunke, H. "A Decision Tree Approach to Graph and Subgraph Isomorphism Detection," *Journal of Pattern Recognition* (32:12) 1999, pp 1979-1980.
- Messmer, B.T., and Bunke, H. "Efficient Subgraph Isomorphism Detection: A Decomposition Approach," *IEEE Transactions on Knowledge and Data Engineering* (12:2) 2000, pp 307-323.
- Opromolla, G. "Facing the Financial Crisis: Bank of Italy's Implementing Regulation on Hedge Funds," *Journal of Investment Compliance* (10:2) 2009, pp 41-44.
- Sadiq, S., Governatori, G., and Namiri, K. "Modeling control objectives for business process compliance," *Business Process Management* 2007, pp 149-164.
- Stijin, G., and Vanthienen, J. "Designing Compliant Business Processes with Obligations and Permissions," *BPM 2006 Workshops, LNCS 4103*, Springer Berlin-Heidelberg, 2006.
- Ullmann, J.R. "An Algorithm for Subgraph Isomorphism," *Journal of the Association for Computing Machinery* (23:1) 1976, pp 31-42.
- Wörzberger, R., Kurpick, T., and Heer, T. "Checking Correctness and Compliance of Integrated Process Models," *10th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing*, 2008, pp. 576-583.

AUTHOR BIOGRAPHIES

Jörg Becker is a full professor for Information Systems at the Department of Information Systems at the University of Münster, Germany. He is managing director of the Department of Information Systems and of the European Research Center for Information Systems (ERCIS). Since 2008, he is vice-rector for strategic planning and quality management of the University of Münster. In 2007, he was invited as a member of the North-Rhine Westphalian Academy of Sciences. He is editor-in-chief for the Information Systems and e-Business Management journal and is editor of several other IS journals. Jörg publishes his scientific papers in leading national and international journals and proceedings. Furthermore, he is editor of several monographs and edited books. His research interests incorporate Business Process Management, Conceptual Modeling, Data Management, Retail Information Systems, ERP Systems, Business Intelligence, and e-Government. Email: becker@ercis.uni-muenster.de

Philipp Bergener is Research Assistant at the Chair for Information Systems and Information Management at the University of Muenster, Germany. He made his diploma in Information Systems at the University of Muenster in 2007. His research focuses on Process Modeling with Semantic Modeling Approaches. Recently, he is primarily focusing on the analysis of such semantic process models, especially concerning weaknesses and compliance. Other research interests include Process Management, Workflows and BI for Public Administrations. He worked in the EU-funded project PICTURE for one and half years. Email: bergener@ercis.uni-muenster.de

Dominic Breuker is working as a research assistant at the Department for Information Systems at the University of Muenster. His research interests include business process management, information modeling and the application of formal methods in these areas. He has published his work on numerous IS conferences including ECIS, WI, HICSS and ACIS. Email: breuker@ercis.uni-muenster.de

Patrick Delfmann is an assistant professor at the Department of Information Systems at the University of Münster. In 2001, he graduated in Information Systems. From 2001 to 2007, he worked as a research assistant at the chair of Prof. Jörg Becker at the University of Münster. In 2006, he received his doctoral degree in Information Systems and took up an assistant professor position in 2007. In early 2011, Patrick was awarded his postdoctoral degree (venia legendi) in Information Systems. He was the coordinator of several research projects, funded by the German Research Foundation (DFG), the German Ministry of Education and Research (BMBF) and the European Union. He teaches at the University of Münster and has been a visiting professor at Moscow State University, at SAP Business School Vienna and Swiss Marketing and Advertising Institute (SAWI). He has published more than 60 scientific papers, many of which have appeared in high ranked national and international proceedings and journals. His main research interests are Conceptual Modeling, Reference Modeling, Conceptual Model Analysis, Semantic Standardization and Business Process Compliance Management. Email: delfmann@ercis.uni-muenster.de

326 **Jörg Becker/Philipp Bergener/, Dominic Breuker, Patrick Delfmann, Mathias Eggert**

Mathias Eggert is a research assistant at the University of Muenster, European Research Center for Information Systems. 2007 he received his bachelor in business information systems from the university of applied science in Ingolstadt. In 2009 he received the master in information systems from the University of Hohenheim. Since 2009 he is a fellow at the Konrad-Adenauer foundation. His research is focused on Business Reporting and Process Compliance Methods for Financial Industries. Email: eggert@ercis.uni-muenster.de