Graph-Based Process Model Matching

Christina Tsagkani(✉)

Department of Informatics and Telecommunications,
National and Kapodistrian University of Athens (NKUA),
Panepistimiopolis, 157 84 Ilisia, Greece
tsagkani@di.uoa.gr

Abstract. Nowadays organizations acquire multiple repositories with process specifications. Organization stakeholders such as business analysts and process designers need to have access and retrieve such information as it is proven that adapting existing business processes in order to meet current business needs is more effective and less error-prone than developing them from scratch. This thesis concentrates on process retrieval and will propose a business process searching mechanism, taking advantage and extending existing graph based matching techniques, with the aim to exploit the knowledge that already exists within an organization.

Keywords: Process model matching · Process model similarity · Graph matching

1 Introduction

The growing orientation in processes of contemporary information systems and service-oriented architectures has led to the existence of repositories with hundreds of process models that are a great source of knowledge. The process retrieval according to user's needs of such repositories has become crucial as it may have multiple applications (e.g. model: reuse, design, merge and conformance).

The problem with traditional search engines is that in most cases they are based on keyword search and text similarity and it is unclear how far search engines are appropriate for process model similarity queries [5]. Thus the aim of this thesis is to propose a technique for process matching that will address both label syntactic and structural metrics.

2 The Research Methodology and Existing Techniques

The proposed mechanism and the related research is driven by the followings: 1. The most natural representation of a business process is to view it as a directed and labeled (attributed) graph where each node represents an activity and each edge a control link between activities. 2. Process discovery uses graph matching techniques in order to identify similar process models.

The similarity aspects as defined in [5] are: Node similarity (similarity of labels and attributes), Structural similarity (e.g. Graph Edit Distance) and Behavioral similarity

F. Fournier and J. Mendling (Eds.): BPM 2014 Workshops, LNBIP 202, pp. 573–577, 2015.
DOI: 10.1007/978-3-319-15895-2_50

(e.g. casual relations between tasks or trace-based semantics). These similarity aspects are aided by the followings: Syntactic similarity, Semantic similarity, Attribute similarity, Type similarity and Contextual similarity.

The Graph matching algorithms used by different techniques can be subdivided into two broad categories based on their output:

- exact matching: defines if two graphs are identical or partially identical.
- inexact matching: determines if two graphs are identical or similar.

Inexact graph matching algorithms can be further distinguished by either finding optimal or sub-optimal (approximate) solutions. More specifically with the former algorithms it is guaranteed to find a solution that matches exactly to the query graph, if it exists. The later algorithms find a solution that is the local minimum of the matching cost and it is not guaranteed to find a solution that matches exactly to the query graph.

The current research is focused on Graph Edit Distance (GED) which is a graph matching paradigm that have been emerged and successfully used in diverse research areas (eg. pattern recognition and data mining). Due to its flexibility it may be applied to all graph types. The idea behind the GED is to define the minimum amount of distortion (using edit operations: insertion, deletion, substitution, join and split) required to transform one graph into another [19].

Reviewing the literature related to graph matching algorithms and GED approaches, algorithms that guarantee optimal result can be found such as [2] that uses the A* algorithm and algorithms that guarantee suboptimal result such as: [17] that suggests a greedy iterative algorithm based on local search that adds a sequence of edit operations calculated on the neighborhood graph matching to the global edit path, [18] that proposes two sub-optimal algorithms (A*beamsearch, A*pathlength), [16] that is based on a quadratic assignment formulation to solve the GED problem, [19] that is based on a linear assignment problem, [8] that uses a linear formulation method to derive lower and upper distance bounds in polynomial time and [20] that is based on local search by optimizing local criteria instead of global.

Besides, there have been many research efforts to address the problem of correspondence between activities of different process models such as: [9] that focuses on process model matching quality, by applying word stemming to labels prior to calculating the similarity scores using, the syntactical technique of Levenshtein [10] and semantically technique of Lin [12, 24] that proposes a framework composed of four types of components (Searchers, Boosters, Selectors and Evaluators), [3] the Triple-S matching technique combines similarity scores of three independent levels: Syntactic level, Semantic level and structural level, [3] the RefMod-Mine/NSCM - N-Ary Semantic Cluster Matching that is based on clustering process model nodes, [3] the RefMod-Mine/ESGM - Extended Semantic Greedy Matching that performs preprocessing to data by using a heuristic filter, semantic word matching using dictionary lookups, a syntactic similarity measure (Levenshtein edit distance [10]) and heuristic grouping based on a set of rules, [4, 5] that considers syntactic and semantic metrics for label similarity and follows a greedy algorithm to search the space of mappings and A* heuristics, [14] that identifies the structural similarity of activities and edges, [11] that identifies Graph Edit Distance by using high level change operations, [6] that focuses on activity labels similarity (syntactic and semantic) and then structural similarity is

measured only when previously there was no perfect match, [13] that uses graph reduction rules and a selective reduce algorithm, while only considers structural similarity (order of activities), [21] that uses features (labels and position of a node into the process structure) in order to find related not similar models, [25] that concentrates on model matching prediction while taking into consideration syntactic, semantic, structural and behavioral aspects and [1] that provides visual queries to return process model matches through label and control-flow matching. Finally some process matching approaches that are based on behavioural similarity are indicatively mentioned in [7, 15, 22, 23].

3 The Proposed Solution

This thesis tackles process matching as a graph matching problem. Taking into consideration that large repositories of business processes are searched against a given query, it is obvious that the perfect match is very rare due to graph variability. Therefore users are well satisfied with results that seem similar to their graph query. Thus the thesis will concentrate on inexact Graph matching algorithms where the challenge is to compute how much two graphs differ or share by transforming process nodes to establish a total mapping using a Graph Edit Distance approach.

More precisely the proposed graph matching technique is going to contribute to the followings:

1. **Similarity metrics used to evaluate mapping distance.** The open research problem related to cost function determination of each edit operation that can be either known a priori or can be defined in a way that favors edit operations that are more likely to take place than others that are infrequent, will be tackled. The basic idea is to use the different similarity aspects as were identified previously in order to define the edit cost functions of operations. It is worthwhile mentioning that most of the existing approaches consider some kind of label matching (syntactic and semantic) in order to judge the similarity of model elements. Thus this thesis is going to estimate edit costs based, not only on label matching but structural matching as well, by taking into consideration aspects such as context, events, data input/output, roles and pre/post conditions.
2. **The algorithm that is used to explore the space of possible mappings.** As process repositories consist of large process models it has been decided to concentrate on providing ways of reducing the search space during the application of the algorithm by applying heuristics, while it will be investigated how the searching mechanism may be aided through the use of graph indexing techniques. As a first step to the current research the Business Process Execution Language (BPEL) has been studied in order to understand how its structural elements are used to describe business processes. It is identified that the type of process elements has a key role in the process matching. Thus it is proposed to use the type accompanied to each node to minimize the search space and instead of comparing each node of the query graph with all nodes of a graph under study, to examine only nodes that are of the same type (e.g. Interaction activities, structured activities, basic activities etc.). Also it is

proposed that the algorithm will follow a prioritization process with hierarchies of types of activities and start pairing/edit-cost measurement of nodes expressing for example interaction activities (because they are considered as a prerequisite and should be included in the result of the algorithm) and if the matching cost does not exceed a given threshold then to continue with other types of activities.

4 Conclusion

A problem that this thesis addresses is how to avail the knowledge from a diversity of process variants that exist in large business process model repositories, as an information resource and use it while creating new processes or optimizing existing ones.

Towards this direction, the current research, while aiming at identifying business process mappings, contributes on proposing a graph-based process matching approach that improves search space complexity and considers aspects that have not been addressed by existing approaches such as context, events, data input/output, and roles.

References

1. Awad, A., Sakr, S., Kunze, M., Weske, M.: Design by selection: A reuse-based approach for business process modeling. In: Jeusfeld, M., Delcambre, L., Ling, T.-W. (eds.) ER 2011. LNCS, vol. 6998, pp. 332–345. Springer, Heidelberg (2011)
2. Bunke, H., Allermann, G.: Inexact graph matching for structural pattern recognition. Pattern Recogn. Lett. **1**(4), 245–253 (1983)
3. Cayoglu, U., et al.: Report: The process model matching contest 2013. In: Lohmann, N., Song, M., Wohed, P. (eds.) BPM 2013 Workshops. LNBIP, vol. 171, pp. 442–464. Springer, Heidelberg (2014)
4. Dijkman, R., Dumas, M., García-Bañuelos, L.: Graph matching algorithms for business process model similarity search. In: Dayal, U., Eder, J., Koehler, J., Reijers, H.A. (eds.) BPM 2009. LNCS, vol. 5701, pp. 48–63. Springer, Heidelberg (2009)
5. Dijkman, R., et al.: Similarity of business process models: Metrics and evaluation. Inf. Syst. **36**(2), 498–516 (2011)
6. Ehrig, M., et al.: Measuring similarity between semantic business process models. In: 4th Asia-Pacific Conference on Conceptual Modelling, vol. 67, pp. 71–80. Australian Computer Society Inc. (2007)
7. Eshuis, R., Grefen, P.W.P.J.: Structural matching of BPEL processes. In: Fifth European Conference on Web Services, 2007, ECOWS 2007, pp. 171–180. IEEE (2007)
8. Justice, D., et al.: A binary linear programming formulation of the graph edit distance. IEEE Trans. Pattern Anal. Mach. Intell. **28**(8), 1200–1214 (2006)
9. Klinkmüller, C., Weber, I., Mendling, J., Leopold, H., Ludwig, A.: Increasing recall of process model matching by improved activity label matching. In: Daniel, F., Wang, J., Weber, B. (eds.) BPM 2013. LNCS, vol. 8094, pp. 211–218. Springer, Heidelberg (2013)
10. Levenshtein, V.I.: Binary codes capable of correcting deletions, insertions and reversals. In: Soviet physics doklady, vol. 10, p. 707 (1966)

11. Li, C., Reichert, M., Wombacher, A.: On measuring process model similarity based on high-level change operations. In: Li, Q., Spaccapietra, S., Yu, E., Olivé, A. (eds.) ER 2008. LNCS, vol. 5231, pp. 248–264. Springer, Heidelberg (2008)
12. Lin, D.: An information-theoretic definition of similarity. In: ICML 1998, Vol. 98, pp. 296–304 (1998)
13. Lu, R., Sadiq, S.K.: On the discovery of preferred work practice through business process variants. In: Parent, C., Schewe, K.-D., Storey, V.C., Thalheim, B. (eds.) ER 2007. LNCS, vol. 4801, pp. 165–180. Springer, Heidelberg (2007)
14. Minor, M., Tartakovski, A., Bergmann, R.: Representation and structure-based similarity assessment for agile workflows. In: Weber, R.O., Richter, M.M. (eds.) ICCBR 2007. LNCS (LNAI), vol. 4626, pp. 224–238. Springer, Heidelberg (2007)
15. Nejati, S., et al.: Matching and merging of state-charts specifications. In: 29th International Conference on Software Engineering, pp. 54–64. IEEE Computer Society (2007)
16. Neuhaus, M., Bunke, H.: A quadratic programming approach to the graph edit distance problem. In: Escolano, F., Vento, M. (eds.) GbRPR. LNCS, vol. 4538, pp. 92–102. Springer, Heidelberg (2007)
17. Neuhaus, M., Bunke, H.: Bridging the Gap Between Graph Edit Distance and Kernel Machines. World Scientific Publishing Co. Inc, River Edge (2007)
18. Neuhaus, M., Riesen, K., Bunke, H.: Fast suboptimal algorithms for the computation of graph edit distance. In: Yeung, D.-Y., Kwok, J.T., Fred, A., Roli, F., de Ridder, D. (eds.) SSPR 2006 and SPR 2006. LNCS, vol. 4109, pp. 163–172. Springer, Heidelberg (2006)
19. Riesen, K., Bunke, H.: Approximate graph edit distance computation by means of bipartite graph matching. Image Vis. Comput. **27**(7), 950–959 (2009)
20. Sorlin, S., Solnon, C.: Reactive tabu search for measuring graph similarity. In: Brun, L., Vento, M. (eds.) GbRPR 2005. LNCS, vol. 3434, pp. 172–182. Springer, Heidelberg (2005)
21. Yan, Z., Dijkman, R., Grefen, P.: Fast business process similarity search with feature-based similarity estimation. In: Meersman, R., Dillon, T.S., Herrero, P. (eds.) OTM 2010. LNCS, vol. 6426, pp. 60–77. Springer, Heidelberg (2010)
22. van der Aalst, W.M., de Medeiros, A.K.A., Weijters, A.: Process equivalence: Comparing two process models based on observed behavior. In: Dustdar, S., Fiadeiro, J.L., Sheth, A.P. (eds.) BPM 2006. LNCS, vol. 4102, pp. 129–144. Springer, Heidelberg (2006)
23. van Dongen, B.F., Dijkman, R., Mendling, J.: Measuring similarity between business process models. In: Bellahsène, Z., Léonard, M. (eds.) CAiSE 2008. LNCS, vol. 5074, pp. 450–464. Springer, Heidelberg (2008)
24. Weidlich, M., Dijkman, R., Mendling, J.: The ICoP framework: Identification of correspondences between process models. In: Pernici, B. (ed.) CAiSE 2010. LNCS, vol. 6051, pp. 483–498. Springer, Heidelberg (2010)
25. Weidlich, M., Sagi, T., Leopold, H., Gal, A., Mendling, J.: Predicting the quality of process model matching. In: Daniel, F., Wang, J., Weber, B. (eds.) BPM 2013. LNCS, vol. 8094, pp. 203–210. Springer, Heidelberg (2013)