Hierarchical Clustering of Process Schemas

Claudia Diamantini, Domenico Potena

Dipartimento di Ingegneria Informatica, Gestionale e dell'Automazione "M. Panti",
Università Politecnica delle Marche - via Brecce Bianche, 60131 Ancona, Italy
{diamantini,potena}@diiga.univpm.it

Abstract. In this work, we focus on the analysis of process schemas in order to extract common substructures. In particular, we represent processes as graphs, and we apply a graph-based hierarchical clustering technique to group similar sub-processes together at different levels of abstraction. We discuss different representation choices of process schemas that lead to different outcomes.

1 Introduction

Process Mining (PM) is the application of inductive techniques to extract general knowledge about business processes from process instances. In state of the art research, instances are traces of running processes recorded in the event logs of ERP, Workflow Management Systems or other enterprise systems, and the goal of PM is to distill a structured process description, from the set of real executions, representing the *process schema* [3]. This mining activity can be exploited for instance to support process mapping activities. In this paper we consider a different process mining task: given a set of process schemas, find groups of similar (sub-) processes. In order to achieve this task, we discuss the application of SUBDUE [1], a hierarchical graph clustering algorithm. Graph clustering techniques have been considered since process schemas have a inherent graph structure, while hierarchical clustering in general, and SUBDUE in particular, allows to account for the inherent abstraction structure typical of processes (from very general macro-processes down to simple activities). Although process schemas can be seen as graphs, the application of SUBDUE requires some choices in terms of how to represent complex flow control structures, like parallel and alternative execution of activities or merging. Sections 2 and 3 discuss different representation choices and their experimental evaluation. Section 4 briefly discusses the results and possible applicative scenarios.

2 Methodology

Given a set of directed graphs $G_i = <N_i, A_i>$ where N_i is the set of nodes and $A_i \subseteq N_i x N_i$ is the set of (possibly labeled) arcs, SUBDUE generates a clustering lattice of typical substructures. In its exact matching version, graphs are iteratively analyzed to discover at each step a cluster of isomorphic substructures. The cluster is then used to

compress the graphs, by substituting to each occurrence of the substructure a single node. The compressed graphs are presented to SUBDUE again, and the process is repeated until no more compression is possible. The output clusters turn out to define a lattice where the clusters are linked if a cluster appears in the definition of another. At each step, the substructure is chosen on the basis of its compression capability, measured by the Minimum Description Length (MDL) heuristics. The description length of a graph is measured by the number of bits needed to represent its adjacency matrix. The algorithm has been successfully applied to analyze structured objects in several domains (see http://ailab.wsu.edu/subdue/) thanks to the flexibility it gives to represent complex objects in terms of mathematical graph structures, and suggesting it as a promising technique to analyze process schemas.

A process schema describes the flow of work performed by a certain number of actors. The kinds of flow include simple sequences of activities (SEQ), and operators used to model parallelization (hereafter called SPLIT) and merging (JOIN) of activities. In particular, a SPLIT-AND means that the end of an activity starts all the linked activities, while in a SPLIT-XOR only one will be executed. Symmetrically, a JOIN-AND indicates that an activity begins when all the previous activities are terminated, while in a JOIN-XOR the completion of a single activity is needed. Figure 1 shows an example of process using some of the described operators in BPMN notation.

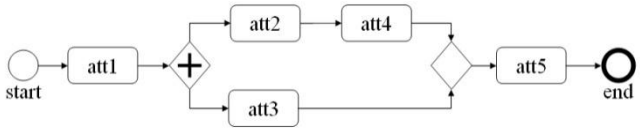

Fig. 1. An example of process schema. Activity *att1* is followed by both *att2* and *att3* (SPLIT-AND), and *att5* is started when *att4* or *att3* are completed (JOIN-XOR).

The application of SUBDUE to business processes requires to perform a mapping from the richer process graph to simpler directed graphs. As we will see, different representation choices may influence the final clustering result. While it seems straightforward to represent the SEQ operator by an arc in the graph, the representation of other operators is not straightforward. We present here three different models, named A, B, and C respectively, and characterized by an increasing level of compactness of the graph, without loss of information. In the A model, any operator is represented by a node called *operator*, which is linked to another node specifying the AND or XOR nature of the operator. In this model join and split are distinguishable by the number of ingoing and outgoing arcs (one outgoing arc and several ingoing arcs for join, the opposite for split). In the B model the node *operator* is replaced by different nodes one for each kind of operator. Finally, the C model simplifies the graph by removing both join and split nodes: since JOIN-XOR and SPLIT-XOR operators represent different alternative executable paths, one for each ingoing (outgoing) activity of a join (split) operator, XOR nodes can be removed by individuating all the possible alternative paths in the process, and generating a graph for each path. In this way, there is no ambiguity about the AND nature of arcs leaving

(entering) a node, so AND nodes can be removed too. Figure 2 shows the representation of the process in Figure 1 with respect to A, B and C models. Note that the three representations hold the same information, and the last produces two compact graphs (one for each xor path). Note also the use of labeled arcs in the C model of Figure 2 to maintain information about domain and range nodes. This is necessary to guarantee the correct interpretation of the final lattice after the compression performed by SUBDUE. It is straightforward to see that these representation strategies can be simply extended to include other BPMN constructs as well (in fact, the first two are directly related to the approach presented in [2]).

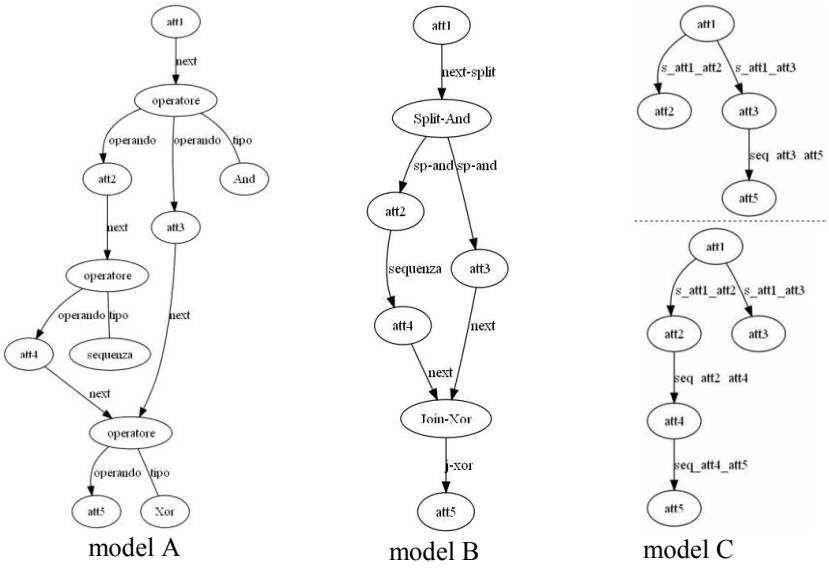

Fig. 2. The representation of the process schema in Figure 1 in conformity with the three proposed models

3 Experimental Evaluation

We experimented the methodology on a set of prototype processes describing e-science activities. In particular, we use a set of data mining processes for the classification task produced in the KDDVM project (http://boole.diiga.univpm.it). Activities are chosen among 21 algorithms of different kind (classification, pre-processing and post-processing) to generate a set of 40 different prototype processes.

In order to evaluate the resulting SUBDUE lattice with different representation strategies and the potentiality of the approach, we introduce some indexes: completeness, representativeness and significance. *Completeness* measures the

number of original graph elements still present in the final lattice[1]. It is expressed as $C = \frac{N_O + A_O}{N_I + A_I}$, where I is the set of input graphs and O is the final lattice. Node completeness is also considered. While completeness measures a quality of the whole lattice, the other indexes allows to individually evaluate each cluster. The *representativeness* of a substructure measures the number of input graphs holding the given substructure at least once. More precisely, representativeness of the substructure S_i is: $R(S_i) = \frac{G(S_i)}{G}$, where $G(S_i)$ is the number of processes holding S_i in graph G. High values of $R(S_i)$ indicate S_i as a typical subprocess. Finally, *significance* is a qualitative index that evaluates the meaning of a cluster with respect to the domain. This index allows us to disregard those clusters that are very representative, but do not contain useful knowledge. In Table 1, we synthetically show results of experimentations in terms of indexes values. In particular, clusters indexes are reported only for high level clusters, which represent the most common substructures. From Table 1, it results that all models are characterized by high completeness, even if C model leads to a slight decrease in the value of such index. The low significance of top level clusters obtained using A model is due to the fact that most frequent substructures are nodes representing individual operators, without references to involved activities. The highest values of representativeness for A model also depends on the high frequency of top level clusters. The C model is that allowing to achieve overall best results, reporting as top level clusters high-frequency substructures that are common in input graphs and are significant in the domain: they are actually knowledge patterns.

Figure 3 shows some of these knowledge patterns. We can see that the most used classification algorithms in the set of data mining processes are BVQ and C4.5. Furthermore, the practice of applying pre-processing algorithms to remove missing values and reduce the dimensionality of datasets emerges as typical patterns. We conclude by noting that SUB_9 and SUB_4 enlighten a not well-formed pattern, since removeMissingValue is performed after LDA. This is not a clustering error, rather it enlighten some problems in input process schemas.

[1] As a matter of fact, during the lattice generation, SUBDUE discards those substructures having low compression capability. This may lead to loose some node or arc.

	A Model	B Model	C Model
Completeness	97%	94%	92%
Nodes Completeness	99%	99%	98%
Representativeness of high level clusters	7%– 67%	8%-31%	8%-40%
Significance of top level clusters	- -	+	+ +

Table 1. Comparison of lattices obtained from graphs represented in accordance with the A, B and C models.

4 Discussion

The paper presents preliminary results about the feasibility of a graph-based clustering approach to recognize similarities among business processes, and to select significant prototypes. In particular, different representation alternatives of a business

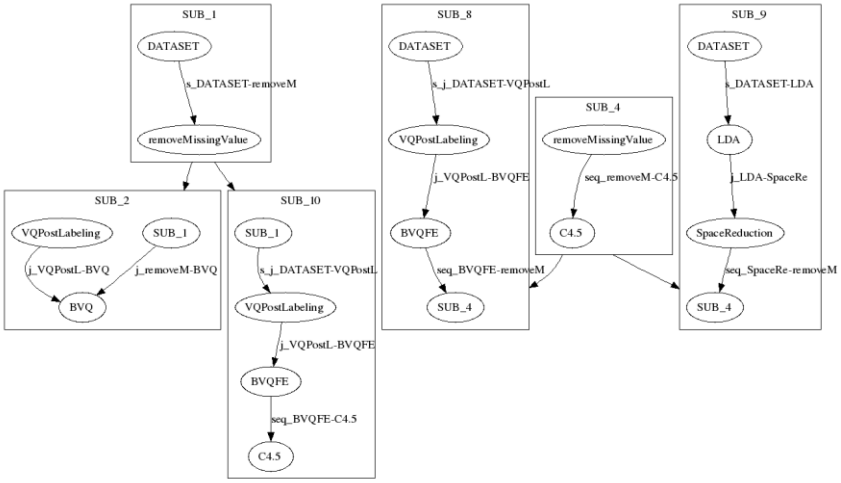

Fig. 3. First two levels of the lattice generated using C model

process for the application of SUBDUE algorithm have been discussed and evaluated. The evaluation on real business processes has been made difficult by the lack of a sufficient number of process schemas, hence we turned to a specialized domain like data mining, exploiting processes automatically generated by an ontology-based composer tool. Nevertheless, this activity allowed to gain useful insights on the method and on the particular domain as well. For instance, from the analysis of the generated lattice we were able to recognize typical patterns of the KDD methodology and we gained insights about some missing or wrong information in the ontology guiding the activity of process generation. The proposed method can find application in a variety of activities related to business process management: first, it can be exploited to individuate similarities and differences in the implementation of certain

processes at different companies, enlightening overlaps, complementarities and heterogeneities, hence supporting enterprise integration at the process level. Second, recurrent common substructures can be exploited to define reference prototype processes and best practices (or common bad practices). Third, the method can be exploited to organize a process repository to enhance search and retrieval. We plan to gather a sufficient number of business processes in order to concretely deal with these applications.

5 Bibliography

1. Jonyer, I., Cook, D. and Holder, L. (2001) Graph-Based Hierarchical Conceptual Clustering, in: Journal of Machine Learning Research, Vol. 2, pp. 19–43.
2. Ouvans, C., Dumas, M., ter Hofstede, A.H.M. and Van der Aalst, W.M.P. (2006) From BPMN Process Models to BPEL Web Services, 2006. International Conference on Web Services, pp.285-292, Chicago, IL, 18-22 Sept. 2006
3. Van der Aalst, W.M.P. and Weijters, A.J.M.M. (2004) Process mining: a research agenda, Computers in Industry 53:231–244.