Towards Automatic Generation of Process Architectures for Process Collections

Rami-Habib Eid-Sabbagh

Hasso Plattner Institute at the University of Potsdam
Prof.-Dr.-Helmert-Str. 2-3
14482 Potsdam, Germany
rami.eidsabbagh@hpi.uni-potsdam.de

Abstract. With the prevalence of business process management in the private and public sector, large process collections are created and shift into focus. To be able to harvest the underlying information, process collections need to be made easily accessible providing intuitive navigation and search.

Process collections are often structured into folders that are labeled along functional, organizational or goal-based lines. As those structures are tediously created manually, they often offer only a single view on the underlying data. However, users use such process collections with different intentions.

This paper presents a generic approach for automatically creating process architectures from unstructured process collection to offer browsing and user centered navigation structures, as well as reduce time of creation. The approach uses the characteristics of clustering algorithms to group processes and label them accordingly. Improvements for further development in the near future will be investigated and outlined as well.

1 Introduction

In recent years BPM has gained momentum in the private and public sector. As a result, process collections of different size, quality, and purpose have emerged. Especially for large collections, the need for an inherent and intuitive structure and navigation is of importance for the retrieval of process models. The knowledge stored in process collections is often not treasured to the best possible extent. The lack of consistent ordering poses strong challenges for the retrieval of process models. According to Baeza-Yates and Ribeiro-Neto [1] providing browsing capabilities on large repositories allows for efficient retrieval of large data sets, especially when users explore information collections without specific intent in mind.

Offering structured overview of business processes in a process collection is one of the aims of process architectures according to Dijkman et al. [2]. They define a process architecture as the relations between processes within a process collection, as well as the guidelines for organizing them. A process architecture shall ensure consistent and integrated process collections; hence enable navigation and easier information retrieval from the process collection.

In well-organized projects, the process elicitation process follows guidelines for structuring process models according to a process architecture. However, in many cases, process architectures have not been developed before the modeling phase. As a result, many process collections are semi- or unstructured.

(Re-) structuring already existing process collections becomes a strenuous manual task, starting with the design of a process architecture. Later, processes need to be classified manually into specific categories. For collections of hundreds or several thousand process models, e.g., SAP Reference Model, Dutch administration, or China CNR Corporation Limited [13], this is not efficient. The manual selection of categories bears possibilities of wrong subjective categorization. Defining crisp and unambiguous rules for classifying process models into categories is rather difficult.

This paper will not focus on defining process architectures in the beginning of business process modeling projects in regard to modeling responsibilities, guidelines, undocumented processes or other issues of process architecture design. It rather presents a generic system design and algorithm architecture along with a concrete example that creates a process architecture based on syntactic similarity of process names. This approach shall provide better navigation through process knowledge, as well as improve efficiency and adequacy over the manual creation of process collection structures. It may even bear the possibility to create process architectures with different focus according to the users' interest.

The paper is structured as follows, Sect. 2 will introduce current research on the structuring of process collections, process architectures and hierarchical clustering, Sect. 4 presents a conceptual system architecture, Sect. 5 sketches an algorithm for creating process architectures, followed by Sect. 6 which will elaborate on future work and improvements of the presented ideas.

2 Related Work

Different approaches to structuring process collections or creating process architectures have been developed and proposed. Most of them are based on manual classification techniques. Weske [17] presents a hierarchy consisting of strategic level, organizational level, operational level and implemented business processes in which business process can be classified according their scope. In contrast to that, Leymann and Roller [7] define a classification of business processes along the dimensions of structure and repetition.

Dijkman et al. [2] present a wide overview of different approaches of designing process architectures. They classify them into five categories; goal-based, action-based, object-based, reference-based and function-based approaches. Each concept shows a different view on a process collection focusing on different aspects of process models.

Scheer et al. [14] design a process architecture consisting of four levels; process engineering, process planning and control, workflow control, and application systems. However, this is rather a classification about the usage of process models in operational activities. Having a different focus, Fettke et al. [4] classify business process reference model approaches according to domain independent and domain dependent characteristics that are functional area and economic activity.

In Smirnov et al. [16] the need for a fast overview of a process's main characteristics is highlighted based on an empirical survey of health insurance workers and validated by BPM consultants. Process architectures have similar aims, e.g. offering information and an overview of the main characteristics of processes in a particular category. Similarly, Melcher and Seese [10] aim to provide more abstract information on process models. They visualize process metrics of process models in a heatmap based on hierarchical clustering methods and a cosine similarity function.

Coming from a different domain but facing similar problems in large multimedia collections Lew et al. [6] emphasize two main necessities, searching for a single item, and browsing and summarizing the information covered by a media collection. Summarizing process information in process architecture categories and providing navigation capabilities are aims of the approach presented in the following sections.

Qiao et al. [11] present a highly effective technique for similarity search of business processes by using clustering algorithms that use structure matching and language modeling. They point out that the clusters found, consist of similar processes as well as provide information about their common characteristics.

Jung et al. [5] propose another technique to find structurally similar process models by adapting a cosine similarity measure to match activity and transition elements of process models. They use an agglomerative clustering algorithm to find similar processes and to create new, or re-engineer business processes in an organization.

Most of the approaches use similarity measure and clustering algorithms to find similar process models or similar process elements focusing on structural [11, 5, 10], semantic aspects within a process model [15], or selected characteristics of process models [10]. However, these approaches only focus on a single process model, or result in displaying only a subset of a process collection.

The reduction of complexity and the visualization of the most important characteristics of process models is the aim of many approaches. Abstraction, as well as providing browsing and summarization of process information can be achieved by creating process architectures. According to Baeza-Yates and Ribeiro-Neto [1] providing browsing capabilities leverages information about the information collection by putting information into context with its environment. So far most approaches for creating process architectures lack automatic support which can be overcome by using hierarchical clustering algorithms for designing process architectures.

3 Hierarchical Clustering

Most of the process architecture styles presented in [2] share hierarchical characteristics to organize their processes whereas they differ according their categorization functions. Hierarchical clustering provides these functionalities and results in a hierarchy according to a selected group of features. In this regard hierarchical clustering seems promising for constructing hierarchical process architectures.

Hierarchical clustering algorithms can be distinguished into agglomerative clustering (bottom-up) and divisive clustering (top-down). Agglomerative clustering algorithms are single-link, average-link, complete-link and centroid-based which differ on their similarity measures. Divisive clustering (top-down) starts with one cluster that iteratively becomes split into smaller clusters. In contrary to agglomerative clustering, divisive clustering is more complex and uses flat clustering algorithms, like k-means clustering in its subroutine [8].

In van Rijsbergen [12] three adequacy requirements for clustering are named, stable clusters for an increasing set of items, tolerability of small errors in the data set, and independence from initial ordering of the items to be clustered. According to van Rijsbergen [12] hierarchical single-link clustering satisfies those requirements. The advantages of this clustering method are efficient search strategies and fast construction of hierarchies in comparison with less efficient search strategies of non-hierarchic structures. Creating hierarchical cluster structures is of high complexity in contrast to K-means clustering and EM-clustering algorithms with low complexity.

Despite that, the use of agglomerative single link clustering (hierarchical clustering) is promising for automatically creating process architectures considering its simplicity and robustness. It fulfills the adequacy requirements making it easily applicable to heterogeneous as well as different process collections. The drawback of exponential computational complexity can be disregarded as creating process architectures is not an everyday task.

4 Conceptual Architecture

Fig. 1 depicts a conceptual system for creating process architectures. It shall provide the flexibility to create different kinds of process architectures by using different similarity measures, cluster algorithms, and different approaches for the labeling of clusters. It consists of four main components, the preprocessor, the clusterer, the label analyzer, and a GUI. The preprocessor takes the process collection as input and formats the attributes, labels, and elements of process models for clustering. The cluster module clusters the preprocessed data according to the selected cluster algorithm. This shall allow exploring the ability of relating clustering algorithms to particular user views. The results of the clustering process

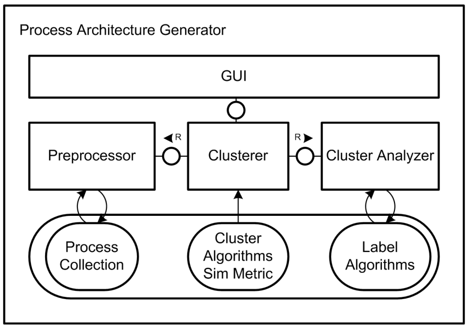


Fig. 1. Process Architecture Generator

are clusters arranged in a hierarchical tree structure. The resulting hierarchy tree diagram is depicted as dendogram [9], see Fig. 4.

After the clustering process, the label analyzer module analyzes each member of a cluster and chooses a label for the cluster considering common characteristics of process models e.g. metadata, labels, or input and output. Algorithms for selecting adequate labels for clusters using semantic approaches from the field of natural language processing will be part of future work.

5 Sketch of Algorithm

The general approach for creating process architectures from process collections, shown in Fig. 2, consists of three steps; preprocessing, clustering, and labeling of clusters, which will be presented along with a concrete example Fig. 3. The approach takes a model collection as input and creates a process architecture with labeled clusters as output. In the preprocessing step, process models and their metadata are cleansed. Metadata values are normalized and missing data is dealt with. String values are converted into numerical values. There are different approaches that can be applied to deal with missing data and string values of metadata. In the example algorithm the preprocessing step consists of extracting the process names from the process models and converting the strings into numerical values by calculating the string edit distance. The output of the step is multidimensional vectors representing the process models. In future research this step could be improved by using natural language processing techniques for dealing with semantics of process names, or labels of control flow elements. For example synonyms could be detected and given the same value. Analyzing the

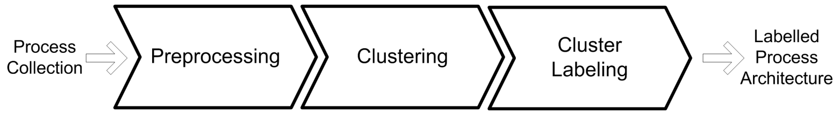


Fig. 2. General algorithm for creating process architectures from process collections

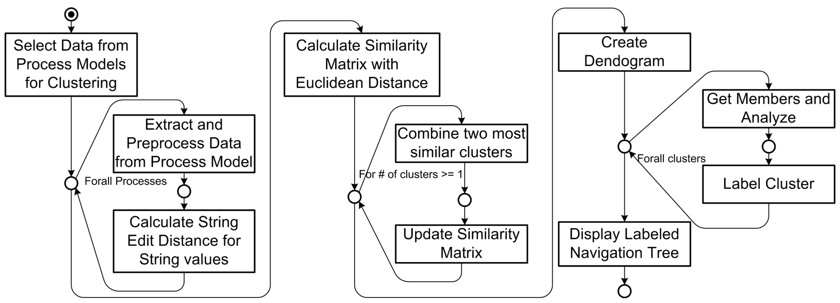

Select Data from Process Models for Clustering
Extract and Preprocess Data from Process Model
Forall Processes
Calculate String Edit Distance for String values
Calculate Similarity Matrix with Euclidean Distance
Combine two most similar clusters
For # of clusters >= 1
Update Similarity Matrix
Create Dendogram
Get Members and Analyze
Forall clusters
Label Cluster
Display Labeled Navigation Tree

Fig. 3. Example Algorithm for creating a process architecture from a process collection

semantic similarity of labels of control flow elements, structural, or behavioral aspects of process models bears many possibilities for exploration. Particular process model elements shall be aligned to styles of process architectures and form the input for generating those automatically; e.g. activities could be used to generate action-based process architectures. The cluster function takes the preprocessed process models, e.g., a multidimensional vector as input as well as the clustering algorithm of interest and a similarity function to calculate the similarity between the different processes in the process collection. The output is a dendogram and a set of clusters.

Function PreprocessProcessModels;

Input *ProcessCollection PC, List SelectionOfMetadata*;

Output *ListPreprocessedModels PM* ;

Function Cluster;

Input *PreprocessedModels PM, ClusterAlgorithm CA, SimFunction SF* ;

Output *Dendogram D, SetClusters C*;

Function ClusterLabeling;

Input *Dendogram D, SetClusters C, AnalysisAlgorithm AA* ;

Output *LabeledDendogram LD, SetOfLabeledClusters LC* ;

Example: Function PreprocessProcessModels;

Input *SAPReferenceModel, Processname*;

Output *multVectors*

Example: Function HierarchicalBottomUpCluster;

Input *multVectors, HierachicalSLBottomUp, EuclideanDistance* ;

Output *(see Fig. 4(a)) Dendogram, Clusters*

Example: Function ClusterLabeling;

Input *Dendogram, Clusters, SimNamePartsAndCountEvents* ;

Output *(see Fig. 4(b)) LabeledDendogram, labeledClusters*

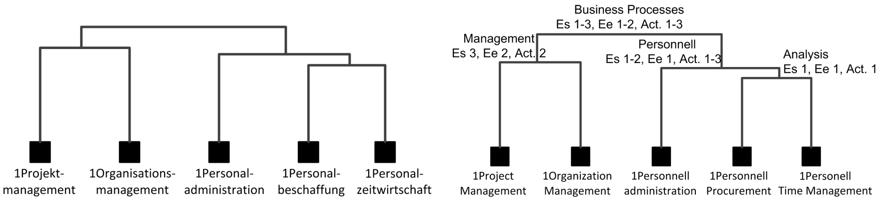
Function General and Example Function Interface Descriptions

The input of the example algorithm is a list of multi-dimensional vectors representing the process models, the bottom-up single-link clustering algorithm and the Euclidean distance function. A hierarchical clustering algorithm is chosen due to the hierarchical nature of most process architectures and the good characteristics of hierarchical clustering algorithms. The Euclidean distance function is used to calculate the similarity matrix for the process models. In future, this step can be realized with more sophisticated similarity functions. Considering each process model as one cluster in the beginning, the hierarchical cluster algorithm will join the two clusters with the minimal distance between any two items p_i and p_j with p_i in cluster i and p_j in cluster j until only one cluster exists containing all other clusters and the inherent process models. After each iteration, the distance matrix must be updated with the similarity value of the newly created cluster. The clustering process results in a dendogram, a hierarchical structure tree that links all the clusters generated.

In the future also flat clustering algorithm can be used with presented framework. The next step defines the labeling process of the clusters. The general algorithm takes the dendogram, the set of clusters and an analysis algorithm as input. Here different strategies that need further elaboration can be applied. E.g. only the input and output labels of each member in the cluster could be extracted and counted.

In the example, the analysis algorithm examines the names of each member of cluster and figures out a word that describes the processes in the cluster. It also counts the number of activities, start events, and end events of each process. The label is put together from a word that is common for each process model in the cluster as well as the range of start events, end events and activities as displayed in Fig. 4(b). In this way context information on the process models in each cluster is provided while browsing. The technique described in the example is rather a simple technique for identifying labels which also leaves room for improvement in the future.

An example output of the clustering algorithm with five process models from the sap reference model collection is depicted in Fig. 4(a) and Fig. 4(b). The implementation of the algorithm has not been fully realized and first results with large process collections can only be presented in the near future.



(a) (b)

Fig. 4. Example of an unlabeled dendogram (a) and a labeled dendogram (b)

Empirical surveys can be used to validate both the automatically created process architectures in regard to their improvement of browsing capabilities and the labels chosen for the different clusters. The architectures can be replicated in process collections and tested with users to investigate the quality of their browsing capabilities. In a similar way, the quality of labels representing the process clusters can be assessed by users in an empirical survey. A current research project, the national process library (npl), offers a suitable use case for this purpose [3]. The process architectures generated could be integrated into the npl and used for browsing. This way browsing capabilities of the process architecture could also be compared to the already existing filter mechanism and search engine in the npl.

6 Conclusion and Future Work

This paper presented a conceptual framework for automatically generating process architectures from process collections. A general and exemplary algorithm as well as the input and output of the different steps were presented to depict the process of generating process architectures for process collections. Suggestions for improvements of the quality of clustering and labeling of clusters were mentioned as future research agenda. Using more sophisticated similarity measures for clustering as well as natural language processing techniques for analyzing semantic information from control flow labels may bear the most potential here. Also, the preprocessing step can be varied in respect to semantics which likely improves results, e.g. only nouns will be extracted. In general the presented approach is very flexible and lays the foundation for future exploration of process collections and the interdependencies of its process models.

References

1. Baeza-Yates, R., Ribeiro-Neto, B.: Modern Information Retrieval. Addison Wesley (1999)
2. Dijkman, R.M., Vanderfeesten, I., Reijers, H.A.: The Road to a Business Process Architecture: An Overview of Approaches and their Use (2011), http://cms.ieis.tue.nl/Beta/Files/WorkingPapers/wp_350.pdf
3. Eid-Sabbagh, R.H., Kunze, M., Weske, M.: An Open Process Model Library. In: Daniel, F., Barkaoui, K., Dustdar, S. (eds.) BPM Workshops. LNCS, vol. 100, pp. 26–38. Springer, Berlin, Heidelberg (2012)
4. Fettke, P., Loos, P.: Classification of reference models: a methodology and its application. Information Systems and e-Business Management 1(1), 35–53 (Jan 2003)
5. Jung, J.y., Bae, J., Liu, L.: Hierarchical Business Process Clustering. 2008 IEEE International Conference on Services Computing 2, 613–616 (2008)
6. Lew, M.S., Sebe, N., Djeraba, C., Jain, R.: Content-based multimedia information retrieval. ACM Transactions on Multimedia Computing, Communications, and Applications 2(1), 1–19 (Feb 2006)
7. Leymann, F., Roller, D.: Production Workflow: Concepts and Techniques. Prentice Hall (2000)
8. Manning, C.D., Raghavan, P., Schütze, H.: Introduction to Information Retrieval. Cambridge University Press (2008)
9. Manning, C.D., Schuetze, H.: Foundations of Statistical Natural Language Processing. The MIT Press (1999)
10. Melcher, J., Seese, D.: Visualization and Clustering of Business Process Collections Based on Process Metric Values. In: 2008 10th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing. pp. 572–575. IEEE (Sep 2008)
11. Qiao, M., Akkiraju, R., Rembert, A.: Towards efficient business process clustering and retrieval: combining language modeling and structure matching. In: Business Process Management. pp. 199–214. LNCS, Springer (2011)
12. van Rijsbergen, C.J.: Information Retrieval. Butterworth (1979)
13. Rosa, L., Arthur, H.M., Efficient, L., Jin, T., Wang, J., La, M.: Efficient and Accurate Retrieval of Business Process Models through Indexing. Computers in Industry (2010)
14. Scheer, A.W., Nüttgens, M., van der Aalst, W., Desel, J., Oberweis, A.: BPM, LNCS, vol. 1806. Springer, Berlin, Heidelberg (Mar 2000)
15. Smirnov, S., Reijers, H., Weske, M.: A semantic approach for business process model abstraction. In: AISE. pp. 497–511. Springer (2011)
16. Smirnov, S., Reijers, H.A., Nugteren, T., Weske, M.: Business process model abstraction: theory and practice. Universitätsverlag Potsdam (2010)
17. Weske, M.: Business Process Management: Concepts, Languages, Architectures. Springer; 1 edition, 1 edn. (2007)