

# A Process Model Collection with Structural Variants and Evaluations

Rimsha Anam<sup>1</sup>, Tahir Muhammad<sup>2</sup>

Department of Computer Science  
COMSATS University Islamabad, Lahore Campus  
Lahore, Pakistan

**Abstract**—Today in the era of the latest technologies, Business Process Management Systems (BPMS) have allowed organizations to build process model repositories which help to maintain the flow of operations in the form of various process models. Business process models are virtual models that can imitate the actual activities of an organization. Searching for semantically similar activities between pairs of process models in a repository is known as Process Model Matching (PMM). From the past few years, PMM has been gaining momentum due to its wide range of applications such as integration of process models, process model clone detection, and process model knowledge discovery. Different types of PMM techniques have been applied on available process model repositories but these repositories contained a limited number of process models. Another notable aspect of PMM is that the existing techniques have not achieved the desired results which questions the effectiveness of process model repositories. To address this problem, the authors of this study have developed a substantial, diverse, and carefully developed process model collection. This process model collection is compared with existing SAP collection to highlight its significance and superiority. Furthermore, the proposed process model collection represents structural variations of example process models which are governed by the defined set of rules. To reflect structural variations between process models of our collection, existing structural similarity approaches such as structural metrics and graph edit distance were applied by using a custom-developed tool. Our proposed process model collection is freely available to the research community which can be used to build new PMM techniques and for assessment of existing PMM techniques.

**Keywords**—Business process modeling; process model collection; Process Model Matching (PMM); structural variants

## I. INTRODUCTION

The process model is a conceptual model that represents various dependencies between the activities of an enterprise. Organizations store their business processes into process model repositories which are considered valuable resources to perform various tasks such as business process improvements, software development requirements, and configuration of Enterprise Resource Planning (ERP) systems [1, 2]. The usefulness of process models has encouraged companies to generate huge collections of process models i.e. a Dutch government organization maintains the dataset of more than six hundred process models [3], a multinational company located in Australian holds a collection of more than five thousand process models, another Chinese factory holds even bigger process model collections [4]. However, such process

model repositories are often proprietary and cannot be shared publicly due to privacy issues.

To improve the handling and reliability of process model collections, more process model repositories with extended features are needed. One of the major challenges is searching a process model effectively and efficiently from a collection [5]. To address this challenge various Process Model Matching (PMM) techniques [6-11] have been proposed that can be used to check similarity between process models. Due to the lack of process model collections with diverse features, the performance of these techniques cannot be evaluated rigorously. At this stage, most of the existing PMM techniques lack empirical evaluation due to the limited availability of larger process model collections[4]. This highlights the need for process model collection that contains a large number of process models that are rich in structural features.

Existing process model collections such as PMMC'15 datasets are freely available which contain only 90 models in total (i.e. university admission = 9, birth registrations = 9 and asset management = 72) [12]. Thus, the PMM techniques evaluated on such collections may not perform well when applied to larger process model collections. PMMC'15 datasets also lack feature diversity because each dataset targets a specific domain. Therefore, in this regard more effort is needed to build a diverse process model collection that can be used for effective evaluation of PMM techniques. As a contribution to solving these challenges, authors have developed a standardized process model collection for rigorous evaluation and improvement of PMM techniques.

The proposed process model collection is generated through a systematic approach and is freely available to the research community and can also be used for comparing different PMM techniques. A total of 750 process models were stored in the proposed process model collection. These process models were offered different shapes, sizes and dimensions, etc. This collection is developed by following the constraint in mind that it will object the ability of a PMM technique to check similarity between process models. The rest of the article is structured as follows: Section II provides a brief background of the terms discussed in this study, Section III provides an overview of different studies related to PMM techniques and process model collections. Section IV explains the protocol used for the development of the proposed process model collection. Different variation types used to generate structural variants of the original process models are discussed in Section V. Custom tool developed to

compute structural metrics of the process models is discussed in Section VI. A detailed analysis of the proposed collection is discussed along with the results in Section VII. In Section VIII, a conclusion is made to highlight the significance and strengths of our proposed process model collection. The last section predicts future dimensions for this study.

## II. BACKGROUND

The process model captures the flow of different activities of an organization. Process model structures can be used to represent the order in which activities of the process model are executed, and this is known as control flow [13]. These process models are stored in process model repositories for current and future use. Searching and adding new process models in repositories are crucial tasks for the maintenance of process model collections. Different PMM techniques are used to avoid duplication and efficient searching of process models in a process model collection. PMM techniques are challenged by structural diversities of process models stored in a collection. To evaluate structural similarities between process models, different structural metrics such as size, density, and complexity between similar process models are used.

These structural metrics were proposed since a decade ago and are widely used and accepted for PMM [14]. By computing and comparing the values of these structural metrics, richness, and diversity of process model collections can also be evaluated. Different PMM techniques lack rigorous evaluation due to the unavailability of large and diverse process model collections for the research community. A more rich and structural diverse process model is required which can be used as a benchmark for meticulous evaluation of PMM techniques.

## III. RELATED WORK

Identification of similar activities between two structural variants of a process model is a crucial task. Traditional search engines perform based on text matching which is not enough in the case of PMM [15]. In the study [15], authors have identified important features of process models that can directly affect various aspects of a process model collection. These features are formally categorized into three categories i.e. 'label feature', 'behavioral feature', and 'structural feature' of a process model [16]. These categories play an important role to address various issues of PMM.

Label feature captures the names of activities of the process model, this group of labels used for activities is called a label feature [17]. Each label can be represented differently for corresponding activities of two similar process models [18]. Due to this fact, the benchmark collection must contain process models with labels having similar semantics but are written in different ways. This can help achieve a more standardized and rigorous evaluation of PMM techniques by detecting different labels with similar meaning.

The behavioral feature of the process model demonstrates a causal relationship between various activities of a process model [19], i.e. indirect edges occurring between different nodes of a process model. Behavioral feature is also an important part of PMM techniques as it can help identify

similarities between two different process models. Comprehensive studies [10, 19] have been conducted to address the issues of process model similarities using label features and behavioral features but similarities based on structural features of process models are comparatively less explored and need more attention [20]. In the study [21], authors have produced different structural variants of a process model without making any change to its semantics. In the study [22], the authors also showed different structural variants of process models. However, only a few structural variants were able to maintain the semantics of the original model.

## IV. PROPOSED PROCESS MODEL COLLECTION

A systematic approach is followed for the development of the proposed process model collection with diverse structural features. It is because, in different studies, authors have recommended employing a systematic approach for process design and process reengineering to reducing human bias [23]. Two types of process model developments were involved in the development of collection: a) *collecting and developing original process models*; b) *producing structural variants of original process models developed in the previous phase*. To avoid human error both phases of model generation were supervised by domain experts who had years of research and teaching experience of business process management and process model repository development. Details of both phases of process model collection are as follows:

### A. Collection and Development of Process Models

Different types of process models were collected from various sources such as books, research papers, technical reports, and other online sources such as example models from Object Management Group (OMG<sup>®</sup>) [24]. No restriction was applied to the selection of sources for original models to target maximum domains. Collection of process models from various domains such as academics, reservation systems, procurement, manufacturing, and payment systems, etc. helped achieve diversity and richness for the proposed process model collection. These collected models were not ready to be stored directly to the process model collection because of the two reasons: 1) these models were in different formats such as images, hand-drawn models, scanned pictures, pdf, videos, etc. 2) these processes were modeled using different modeling languages i.e. Petri-Nets, EPC, BPMN, YAWL, etc. So all of these models were redesigned in Business Process Modeling Notation (BPMN) using a modeling tool called CAMUNDA Modeler [25] which is a lightweight and open-source software that supports BPMN standards. The selection of the tool was inspired by two causes: 1) it comes with various modeling guidelines and generates error-free models; 2) it allows users to export process models in different file formats such as PDF and XML. While modeling these processes, few changes were made to ensure diversity between different models. Furthermore, the process models went through manual testing for logical errors and were improved according to widely accepted modeling guidelines [26]. These modeling guidelines allowed the addition or removal of few elements to ensure that the produced model conformant with existing standards of process modeling. For example, a couple of process models

have *OR-split* with a missing corresponding *OR-join*. It was against BPMN modeling guidelines, according to which each *OR-split* must be accompanied by an *OR-join*. At the end of *Phase I*, a total of 150 models were generated and stored into process model collection.

**B. Generating Process Model Variants**

The goal of this phase was to improve process model collection by adding more process models to the collection that offer distinct structural features. The focus of this study is to create structural diversity between similar types of process models. Considering this fact proposed process model collection was extended by adding 600 structural variants of the original 150 process models. The major cause for the introduction of structural diversity between similar types of process models is that it can help evaluate various PMM techniques rigorously.

Four different types of structural variants were generated from each original process model. These four structural types were inspired by traditional similarity metrics of process models discussed in studies from [21, 22, 27]. From these studies, variation patterns that do not change the semantics of the original process models were chosen for the classification of process model variants of the proposed collection. Classification of structural variants is explained in detail in Section V.

**C. Process Model Collection Dimensions**

The proposed process model collection contains 750 process models out of which 150 are original models and 600 are different structural variants of the original 150 process models. Table I shows the number of process models along with their types. These process models were designed using CAMUNDA Modeler [25] as a modeling tool, and the modeling language used was the BPMN.

TABLE I. STATISTICS OF PROPOSED COLLECTION

1.	Original Models	150
2.	SVT-1	150
3.	SVT-2	150
4.	SVT-3	150
5.	SVT-4	150
Total		750

To validate the strengths of the proposed collection, analysis of the proposed collection was performed. The analysis was conducted by comparing the proposed collection with existing ones and its internal consistency was evaluated by comparing all four variants with original 150 process models.

**V. CLASSIFICATION OF STRUCTURAL VARIATION TYPES**

Classification of structural variations of the original process model is performed as follows: 1) In first variation different structural variants of the original process model were generated by through defined traces. 2) The second variation involves the addition of activities to the original process models. 3) The third variation adds edges to modify control follow of the process model. 4) Structural variants were produced by swapping common activities which are loosely bounded, without changing the semantics of the original process model.

To explain these four classes: structural variation type 1 (SVT-1), structural variation type 2 (SVT-2), structural variation type 3 (SVT-3), and structural variation type 4 (SVT-4), different structural variants of an original process are generated. The sample process model selected to demonstrate the generation of structural variants is a Facebook Co. process model for registration of new candidates into the system. The process model contains 12 activities, 6 gateways, and 2 events as shown in Fig. 1. All possible traces of the Facebook Co. process model are generated which are shown in Fig. 2.

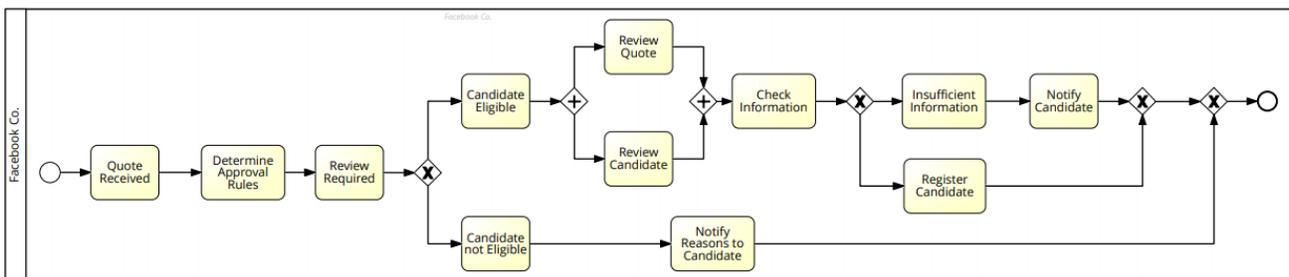


Fig 1. Example of Original Process Model.

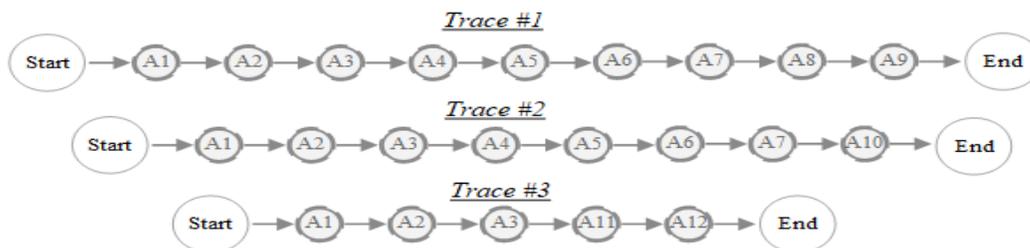


Fig 2. Traces of Original Process Model.

A. SVT-1

In SVT-1, sequential activities can be executed in parallel by defining the traces of the process model. The execution patterns demonstrate the execution order of the activities of the process model, also known as traces of a process model. First, all possible execution patterns of the process model are defined without considering any control block. Activities occurring before and after of a control block are considered as common activities in all traces of the process model. After marking the common activities, the sequential order of activities within any control block is arranged in a way to observe parallel execution through AND gateway.

Each process model has control flow which defines how the execution of activities can be conducted for a process model. The control flow is modified with the help of different types of gateways i.e. AND, OR, and XOR. Through these gateways' activities are executed in parallel or exclusively. In parallel execution of activities, AND-split is used along with AND-join to synchronize the flow of execution. In mutually exclusive execution of activities XOR-split is used along with XOR-join to merge alternative flows. In mutually exclusive flow of execution, the selection of activity that needs to be executed depends upon an external event or data. By making changes to control flow, different structural variants were produced. A structural variant generated by SVT-1 can be transformed back to its original process model by developing a graph of its execution traces. Table II explains design patterns and conditions for the generation of process model variants according to SVT-1.

B. SVT-2

In SVT-2 one or more activities are added to a process model to make a structural change. However, this change in structure will not modify the process model semantically. To add one or more activities first, understand the description of the original model by exploring all of its activities, then the activity is added between two succeeding activities by following the design pattern discussed in Table III.

According to SVT-2 new activity can only be introduced between a pair of activities that are common in traces, and if they are not depending on each other. By following SVT-2 variants of the original models were generated by the addition of new activities, but the original model remained semantically the same. However, if we add new activity between two activities that are closely bounded than it will change the behavior of the original model semantically.

TABLE II. VARIATIONS WITH TRACES (SVT-1)

<b>Description</b>	
Activities without any strict execution pattern can be executed in parallel.	
<b>Design Pattern</b>	
1.	All possible traces of an original process model are defined.
2.	Activities that occur as pre and post activities of the gateway in the traces are called common activities.
3.	Activities within the control block which were executed in a sequence are executed in parallel by introducing AND gateway.
4.	By following steps 2 and 3, all common activities in traces are connected with the activities of the control block by using gateways of the original process model.
5.	All traces should be operated independently.
<b>Condition</b>	
Activities are not tightly bound for sequential execution. Traces are defined without considering any gateways.	
<b>Graphical Representation</b>	
Fig. 3.	

C. SVT-3

Each process model has its control flow which is defined in the form of vertices and edges. SVT-3 deals with the addition of the control edge structure to the original process model by considering pre and post conditions of the activities. The process models observe four basic types of control flow mechanisms i.e. sequential control flow structure, parallel control flow structure (AND-Gateways), choice control flow structure (XOR-Gateways), and looping control flow structure. For sequential control flow all activities are executed in a sequence and succeeding activity is executed after executing its previous one. In parallel execution, AND gateway is used to execute two activities simultaneously while on the other hand XOR gateway is used to offer an exclusive execution among one of the two succeeding activities. In looping structures activities are executed repeatedly in a loop. According to SVT-3 new control edge can be added to all of these four control flow structures by considering pre and post conditions of the activities. According to the study [22], structure variations are identified based on the change primitives such as the addition and modification of nodes and edges in the original process model through the design patterns mentioned in Table IV. However, the addition or deletion of a new control edge does not effects the semantics of the process model [21, 22].

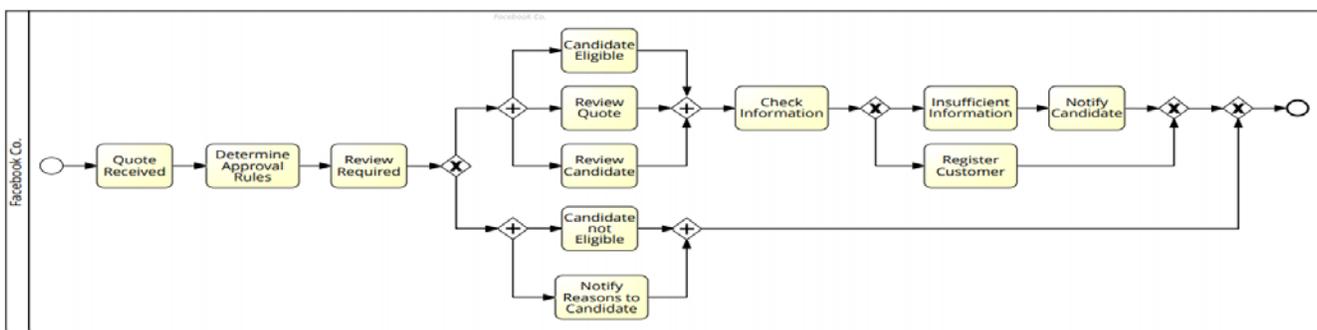


Fig 3. SVT-1 of Original Process Model.

TABLE III. VARIATIONS WITH ADDITION OF ACTIVITIES (SVT-2)

<b>Description</b>
One or more activities are appended with the original process model without effecting its semantics.
<b>Design Pattern</b>
<ol style="list-style-type: none"> <li>1. An activity can be introduced in the middle of two succeeding activities.</li> <li>2. The common activities of the traces defined in (SVT-1), can accommodate one or more new activities.</li> <li>3. New activity could provide a structural variation in the process model, but it will remain semantically the same.</li> </ol>
<b>Condition</b>
One or more activities can only be added when existing activities are not rigorously bound.
<b>Graphical Representation</b>
Fig. 4.

TABLE IV. VARIATIONS WITH MODIFICATION OF CONTROL EDGE (SVT-3)

<b>Description</b>
An addition of control dependency can be added among existing control flow.
<b>Design Pattern</b>
<ol style="list-style-type: none"> <li>1. Control dependency can be added among existing control flow by evaluating the pre and post conditions of the activities.</li> <li>2. The control edge will add up new the condition in the control flow.</li> <li>3. The new condition should be consistent with the existing control block.</li> </ol>
<b>Condition</b>
SVT-3 can only be applied after validating pre and post conditions of control flows in the process model.
<b>Graphical Representation</b>
Fig. 5.

D. SVT-4

According to the SVT-4 order of the common activities in the traces is modified to generate a structural variant of the original process model. First, the execution pattern of the original process model is identified so that activities outside control blocks can be recognized. Another important thing to be considered before doing swapping between activities is that the targeted activities should not be closely bounded. If the target activities are closely bounded than these changes can change the semantics of the variant concerning the original process model. Design pattern rules for SVT-4 are mentioned in Table V. along with the conditions. In Study [21] authors suggest that even after swapping of activities, the semantics will remain the same.

TABLE V. VARIATIONS BY SWAPPING ACTIVITIES (SVT-4)

<b>Description</b>
One or more common activities in traces can be swap with each other.
<b>Design Pattern</b>
<ol style="list-style-type: none"> <li>1. Used the execution traces discussed in SVT-1.</li> <li>2. One or more common activities can only be swapped if they are in a succeeding execution order.</li> <li>3. Swapping can be possible among common activities of the traces.</li> </ol>
<b>Condition</b>
Swapping cannot be applied to activities with strict execution order.
<b>Graphical Representation</b>
Fig. 6.

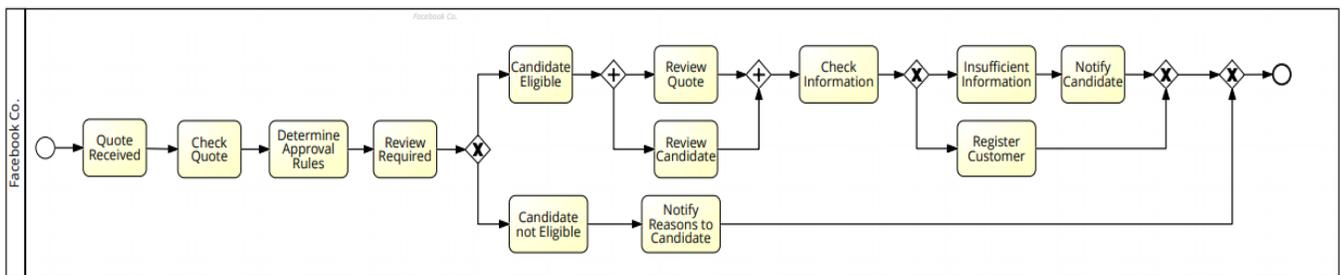


Fig 4. SVT-2 of Original Process Model.

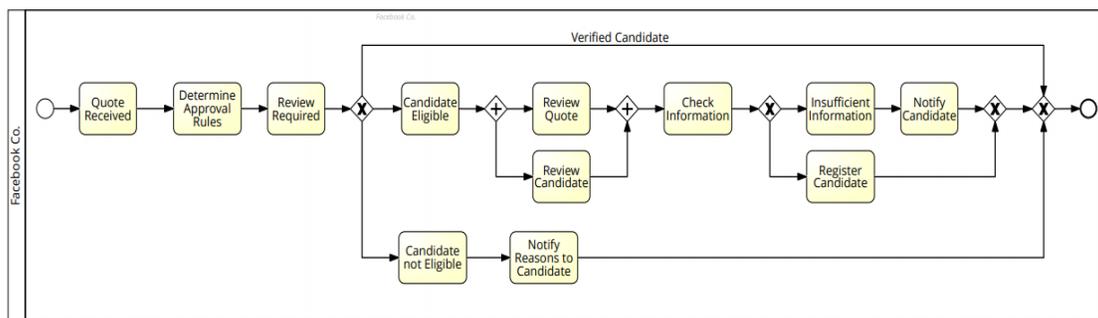


Fig 5. SVT-3 of Original Process Model.

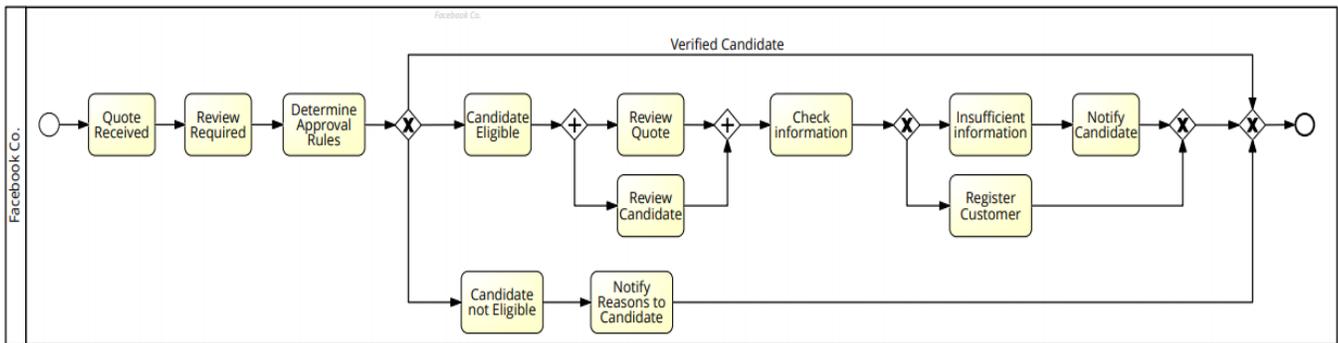


Fig 6. SVT-4 of Original Process Model.

## VI. PROCESS MODEL MATCHING TOOL

A custom tool was developed to compute different structural metrics and GED of the process models stored in the proposed collection. The proposed tool is a desktop application that runs on windows operating system. Java programming language was used for the development of this tool. Fig. 7 shows the functional behavior of the proposed process model matching tool. To perform different types of computations, first all of the process models were stored into XML format by using CAMUNDA modeler. XML files are processed to extract information such as process name and lanes information, Events and task information, control flow information along with their sequence mappings. This information was stored in the form of variables and these

variables were passed to different functions to compute metrics. By mapping various execution sequences business process graphs were produced. Separate functions were written to compute graph edit distance of the process models.

Similarly, BPGs of the process model were generated using extracted information of the process models, and these BPGs were used as input to compute GED. Different GED operations were performed i.e. *Node Insertion or Deletion (SN)*, *Edge Insertion or Deletion (SE)*, *Node Substitution (SB)* on these BPGs, through which values of GED and similarity were computed. Structural metrics and all other results generated by the proposed tool are also uploaded and are available online for the research community [28].

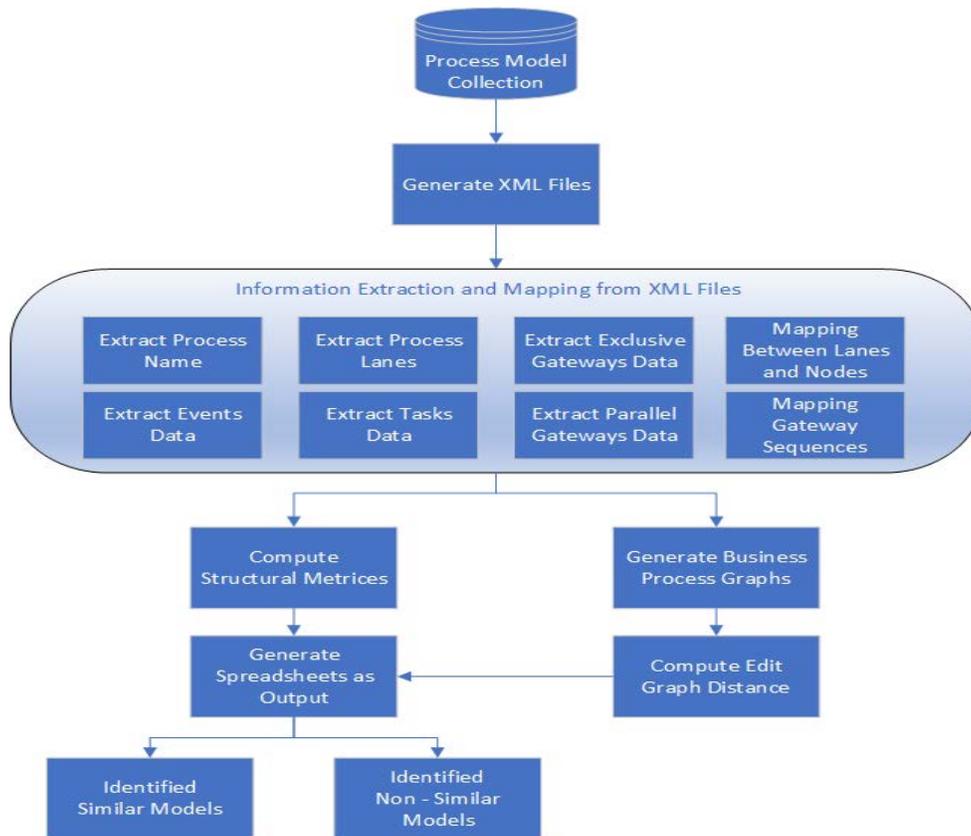


Fig 7. Functional Behavior of Custom Tool for PMM.

## VII. ANALYSIS AND RESULTS

The proposed process model consists of structurally diverse models that are rarely found in existing process model collections. To evaluate the structural diversity of the proposed collection two methods were used first, it was compared with existing SAP's process model collection using widely accepted structural metrics. It should be noted that SAP's process model collection is not openly available to the public so we relied on the results published in previous studies [14, 29].

Secondly, similarity between the original process model and its structural variants is computed by using graph edit distance (GED). The purpose of this computation is to highlight the fact that produced structural variants are semantically similar to their original process model and show less similarity with the rest of the process models in the proposed process model collection.

### A. Comparison of Structural Metrics

Each process model has structural features that can reveal various dimensions of the process model i.e. relationship between various nodes, number of arcs (density), the sequence of execution, etc. There are different types of structural features which can be used for evaluation of our proposed collection. However, to perform a standardized comparison we have chosen widely accepted and most discussed structural metrics proposed by *Mendlings* [14] for process models similarity evaluation. These metrics were designed for various domains such as network analysis, software measurement, and business process models. In this study [14], 26 structural metrics are considered out of which 15 metrics are associated with the size of a process model, and rest are used to express other aspects such as density, sequentiality, separability, and cyclicity of the process model. To compute these structural metrics, a custom tool was developed which is discussed previously in Section VI.

TABLE VI. COMPARISON OF STRUCTURAL METRICS

Structural Metrics	SAP Collection		Proposed Collection	
	Mean	Std.	Mean	Std.
Total Number of Nodes	18.74	20.74	21	8.3
Total Number of Functions	3.81	4.03	19	8.3
Total Number of OR Gateway	1.81	1.08	0	0
Total Number of XOR Gateway	2.78	1.95	2.8	1.5
Number of AND Gateway	2.75	2.18	2.1	2.8
Density of Process Models	0.09	0.08	0.01	0.02
Average Degree of Connectors	1.46	3.3	3.1	0.8

Due to space limitation, we have presented the comparison values of a few structural metrics as shown in Table VI and rest are made available online [28]. It is important to mention here that SAP's process models were modeled in EPC, while for proposed collection BPMN is used as a modeling language. EPC and BPMN are two different modeling languages with different modeling rules. In EPC, each function has one pre-event and one post-event while in BPMN there is no such specific restriction. For instance, a linear process model in EPC with a size of 14 contains 6 functions and 7 events while on the other hand a BPMN process model with size 14 will have 12 activities with 2 events (*Start and End Events*). Considering this fact that there are a smaller number of intermediate events in BPMN than EPC, the process models stored in the proposed collection are more functional as compared to process models in SAP's collection.

In Table VI, the number of nodes that represent the size of the process models has a value of 21 for the proposed collection which is a little bit higher than SAP's collection value i.e. 18.74. This shows that the size of the process models in the proposed collection is almost similar to the ones in SAP's collection. However, considering the richness and functionality of the process models, the proposed collection offers an average value of 19 whereas SAP's collection has a value of 3.81. This concludes that when it comes to functionality the size of process models store in the proposed collection is greater than process models available in SAP's collection. It is also mentioned that the proposed collection does not contain any process model with OR gateway. This was established by following state of the art process modeling guidelines which recommend the use of XOR gateway instead of OR gateway [30]. Due to the absence of OR gateway in the process models its value for the proposed collection is zero. Another structural metric of a process model is density, which indicates the ratio of arcs to the maximum number of arcs that can exist between the same number of nodes. A higher value of density for a process model indicates more arcs between nodes of a process model. It is also a recognized fact, that many of the natural graphs follow a 'power law' according to which models with greater number of nodes have less number of arcs which represent low density [14]. Hence the proposed collection has almost a similar number of arcs as compared to SAP's collection but in terms of functionality and diversity the proposed process model collection is richer and larger (the number of nodes is greater).

Another structural metric is the average degree of connectors, and its higher value for the proposed collection highlights the existence of more logical control flow gateways in process models which indicates structural richness of proposed collection as compared to SAP's collection.

### B. Graph Edit Distance (GED)

In continuance with evaluating strengths of the proposed process model collection, we decided to perform mapping of structural variants to the original process model by using GED [15]. The minimum number of graph-edit operations that are required to get from one graph to another is called GED. It helps to discover the level of structural similarity between original process models and their variants. With the help of similarity values between process models we can verify the

correctness of the proposed collection. To compute the GED of a process model, first the model was converted into Business Process Graph (BPG) [31]. Three types of graph-edit operations that can be performed on these BPGs to compute structural similarity. 1) *Node Insertion or Deletion (SN)*. 2) *Edge Insertion or Deletion (SE)*. 3) *Node Substitution (SB)* [15].

After performing these operations on BPGs, the value of graph-edit distance was computed between each original process model and its four structural variants by using a custom tool discussed in Section VI. A reference process model i.e. P2 was taken to compute overall similarity between all process models.

Higher values of GED indicate minimum structural similarity among a pair of process models. To explain, GED and structural similarity values of proposed collection we have picked 10 random computations as shown in Table VII. The computed values of similarity can occur between 1 and 0 where 1 indicates maximum and 0 means minimum similarity.

After computations, it was observed that each original process model shows maximum similarity with all of its four structural variants. However, when it comes to the general comparison of one process model with other original process models than the similarity values are decreased. This shows that the four structural variants generated against each original process for the proposed collection are semantically similar but structurally different, which makes the proposed collection optimal for the evaluation of PMM techniques.

TABLE VII. COMPUTATIONS OF GED AND SIMILARITY

Query Model	GED	Similarity	Document Model
P2 (Original)	0.00	1.00	P2.0 (Original)
	17	0.89	P2.1 (SVT-1)
	4	0.95	P2.2 (SVT-2)
	1	0.99	P2.3 (SVT-2)
	0	0.98	P2.4 (SVT-4)
	42	0.621	P10.2 (SVT-2)
	35	0.6	P3.4 (SVT-4)
	41	0.65	P8 (Original)
	42	0.67	P30.3 (SVT-3)
	42	0.7	P117.1 (SVT-1)
28	0.76	P4.1 (SVT-1)	

### VIII. CONCLUSIONS

A process model collection is proposed in this study, which consists of 750 process models with distinct structural features. To generate different structural variants of process models, a systematic approach was deployed. To demonstrate strengths of the proposed collection over existing collections a detailed analysis was performed. Two state of the art approaches were employed for rigorous analysis of the proposed process model collection i.e. structural metrics and graph edit distance. First, different structural metrics for both collections were computed through a custom-developed tool. By comparing the values of these metrics, it was found that

the proposed collection contains process models with more diverse and activity-rich. It was also found that process models in the proposed collection are strictly aligned with the modeling guidelines and are comparatively well structured to SAP process models. Secondly the graph edit distance of process models is computed to evaluate the similarity between process models of the proposed collection. The values of similarity between variants in results showed that the proposed collection is consistent with has the potential to be used as a tool to evaluate various process model matching techniques.

### IX. FUTURE WORK

In the future, new PMM techniques can be developed, or existing PMM techniques can be improved by deploying them on the proposed process model collection. Combing structural similarity features along with label and behavioral features can also be performed which may produce some interesting results. It is also mentioned that machine learning could be employed for PMM or evaluation of existing PMM techniques. Another dimension to this study could be the comparison between different similarity features such as understanding the pros and cons of structural similarities with label and behavioral features.

### REFERENCES

- [1] Kuss, E., H. Leopold, C. Meilicke, and H. Stuckenschmidt. Ranking-based evaluation of process model matching. in OTM Confederated International Conferences" On the Move to Meaningful Internet Systems". 2017. Springer.
- [2] Dumas, M., M. La Rosa, J. Mendling, and H.A. Reijers, Fundamentals of business process management. Vol. 1. 2013: Springer.
- [3] Kunze, M., M. Weidlich, and M. Weske, Querying process models by behavior inclusion. *Software & Systems Modeling*, 2015. 14(3): p. 1105-1125.
- [4] Yan, Z., R. Dijkman, and P. Grefen, Generating process model collections. *Software & Systems Modeling*, 2017. 16(4): p. 979-995.
- [5] Tax, N., N. Sidorova, and W.M. van der Aalst, Discovering more precise process models from event logs by filtering out chaotic activities. *Journal of Intelligent Information Systems*, 2019. 52(1): p. 107-139.
- [6] Xue, X., Compact memetic algorithm-based process model matching. *Soft Computing*, 2019. 23(13): p. 5249-5257.
- [7] Huang, Y., W. Li, Z. Liang, Y. Xue, and X. Wang, Efficient business process consolidation: combining topic features with structure matching. *Soft Computing*, 2018. 22(2): p. 645-657.
- [8] Huang, H., Z. Lu, R. Peng, Z. Feng, X. Xuan, P.C. Hung, and S.-C. Huang, Efficiently querying large process model repositories in smart city cloud workflow systems based on quantitative ordering relations. *Information Sciences*, 2019. 495: p. 100-115.
- [9] Shahzad, K., I. Pervaz, and A. Nawab. WordNet based Semantic Similarity Measures for Process Model Matching. in BIR Workshops. 2018.
- [10] Shahzad, K., S. Kanwal, and K. Malik. Process Model Matching with Word Embeddings. in International Conference on Intelligent Technologies and Applications. 2018. Springer.
- [11] Baier, T., C. Di Ciccio, J. Mendling, and M. Weske, Matching events and activities by integrating behavioral aspects and label analysis. *Software & Systems Modeling*, 2018. 17(2): p. 573-598.
- [12] Pergl, R., E. Babkin, R. Lock, P. Malyzhenkov, and V. Merunka, Enterprise and Organizational Modeling and Simulation: 14th International Workshop, EOMAS 2018, Held at CAiSE 2018, Tallinn, Estonia, June 11–12, 2018, Selected Papers. Vol. 332. 2018: Springer.
- [13] Augusto, A., R. Conforti, M. Dumas, M. La Rosa, and A. Polyvyanyy, Split miner: automated discovery of accurate and simple business

- process models from event logs. *Knowledge and Information Systems*, 2019. 59(2): p. 251-284.
- [14] Mendling, J., Metrics for process models: empirical foundations of verification, error prediction, and guidelines for correctness. Vol. 6. 2008: Springer Science & Business Media.
- [15] Dijkman, R., M. Dumas, B. Van Dongen, R. Käärrik, and J. Mendling, Similarity of business process models: Metrics and evaluation. *Information Systems*, 2011. 36(2): p. 498-516.
- [16] Carmona, J., B. van Dongen, A. Solti, and M. Weidlich, *Conformance Checking*. 2018: Springer.
- [17] Leopold, H., H. van der Aa, J. Offenbergh, and H.A. Reijers, Using Hidden Markov Models for the accurate linguistic analysis of process model activity labels. *Information Systems*, 2019. 83: p. 30-39.
- [18] Zeng, Q., X. Tang, W. Ni, H. Duan, C. Li, and N. Xie, Missing Procedural Texts Repairing Based on Process Model and Activity Description Templates. *IEEE Access*, 2020. 8: p. 12999-13010.
- [19] Liu, C., Q. Zeng, H. Duan, S. Gao, and C. Zhou, Towards comprehensive support for business process behavior similarity measure. *IEICE TRANSACTIONS on Information and Systems*, 2019. 102(3): p. 588-597.
- [20] Kim, K.P., Functional Integration with Process Mining and Process Analyzing for Structural and Behavioral Properness Validation of Processes Discovered from Event Log Datasets. *Applied Sciences*, 2020. 10(4): p. 1493.
- [21] Weber, B., M. Reichert, and S. Rinderle-Ma, Change patterns and change support features—enhancing flexibility in process-aware information systems. *Data & knowledge engineering*, 2008. 66(3): p. 438-466.
- [22] Becker, M. and R. Laue, A comparative survey of business process similarity measures. *Computers in Industry*, 2012. 63(2): p. 148-167.
- [23] Banica, B. and L. Patricio. *Service Design for Business Process Reengineering*. in *International Conference on Exploring Services Science*. 2020. Springer.
- [24] “Informative Documents OMG®” October 2019. [Online]. Available: <https://www.omg.org/spec/BPMN/>.
- [25] “Camunda Modeler” October 2019. [Online]. Available: <https://camunda.com/>.
- [26] Corradini, F., A. Ferrari, F. Fornari, S. Gnesi, A. Polini, B. Re, and G.O. Spagnolo, A guidelines framework for understandable BPMN models. *Data & Knowledge Engineering*, 2018. 113: p. 129-154.
- [27] Yan, Z., R. Dijkman, and P. Grefen. Generating synthetic process model collections with properties of labeled real-life models. in *Asia-Pacific Conference on Business Process Management*. 2014. Springer.
- [28] “Proposed Process Model Collection and Structural Metrics,” March 2020. [Online]. Available: <https://bit.ly/3bYyH37>.
- [29] Mendling, J., H. Verbeek, B.F. van Dongen, W.M. van der Aalst, and G. Neumann, Detection and prediction of errors in EPCs of the SAP reference model. *Data & Knowledge Engineering*, 2008. 64(1): p. 312-329.
- [30] Mendling, J., H.A. Reijers, and W.M. van der Aalst, Seven process modeling guidelines (7PMG). *Information and Software Technology*, 2010. 52(2): p. 127-136.
- [31] Dumas, M., L. García-Bañuelos, M. La Rosa, and R. Uba, Fast detection of exact clones in business process model repositories. *Information Systems*, 2013. 38(4): p. 619-633.