SRTTU

Journal of Electrical and Computer Engineering Innovations
(JECEI)
Journal homepage: http://www.jecei.sru.ac.ir

JECEI

Research paper

A Multi-Aspect Semi-Automated Service Identification Method

S. Hekmat[1], *S. Parsa*[2,*], *B. Vaziri*[1]

[1]*Department of Computer Engineering, Central Tehran Branch, Islamic Azad University, Tehran, Iran.*

[2]*Department of Computer Engineering, Iran University of Science and Technology, Tehran, Iran.*

Article Info

Article History:
Received 05 December 2021
Reviewed 15 January 2022
Revised 17 February 2022
Accepted 25 April 2022

Keywords:
Service identification
Business process model
Process mining
Service-oriented architecture
Goal model
Data model

*Corresponding Author's Email Address: *parsa@iust.ac.ir*

Abstract

Background and Objectives: Several service identification methods have been proposed to identify services using business process-based strategy. However, these methods are still not accurate enough and adequately automated and thus need improvements. The present study addresses this gap by proposing a new semi-automated combinational method that applies process mining techniques and simultaneously considers different aspects of the business domain (e.g., goal and data). We argue that this method facilitates service identification more comprehensively and accurately and helps enhance organizational performance and lower cost structure.

Methods: Our method includes three Phases. In the first phase, the system log is inspected, and the running business process is extracted using process mining techniques. After validating this model, we create a goal and data model in the next phase. In the third phase, we establish connections between the introduced models by defining some matrices. These connections are of two types: structural and conceptual. Finally, we propose a couple of algorithms that lead to the identification of services.

Results: We evaluate the utility of our proposed method by conducting a case study and using the experts' opinions from different perspectives as follows: (1) assessing the accuracy and reusability of the identified services, (2) appraising the efficiency of employing this method in more complex processes, (3) calculating the cohesion to coupling ratio, and (4) assessing the performance of the method and other service quality measures. The results indicate that the average accuracy of this method is about 12 % higher than the previously identified methods for both simple and complex processes. Additionally, it empirically proves that using the process mining techniques improves the service identification considerably (8%). Moreover, according to the experts' opinions, the combination of goal and data model and process mining has increased the performance by 8%. In comparison, cohesion to coupling ratio demonstrated a 7% increase compared to other methods. In sum, we conclude that this method is an advanced and reliable way of service identification regardless of the process size and the complexity.

Conclusion: The findings reveal that considering different aspects of business processes together and using process mining techniques improves the ratio of cohesion to coupling and accuracy of the identified services. Adherence to this approach enables companies to mine their business processes, modify them, and quickly identify services with higher performance. Besides, using this method provides a semi-automated and more effective way of service identification.

Introduction

In each organization, several business processes exist that need to be reviewed and changed continuously to comply with the organization's goals [1], [2]. Using the service-oriented architecture (SOA) is very helpful for the

Doi: 10.22061/JECEI.2022.8151.526

required changes due to creating reusable and easy-to-change business services.

The first step in moving toward SOA is service identification. There are different strategies for this purpose, such as business processes analysis. Many methods have adopted this strategy and identified services by decomposing business process models defined by experts in the design phase. Recognizing potential services in the strategy relies on understanding and identifying relations between activities, especially those with loose coupling and high cohesion dependencies in the business process models [3]-[5]. Using models in the strategy has caused simplicity and understandability. Besides, the output of the identified services will address the business needs [6]. Nevertheless, service identification based on the strategy in the run-time and dynamic environment might face some challenges.

First, the major problem with relying on predefined BPMs to identify services is that such models are not always available, especially for legacy software systems [7]. In addition, there are some differences in most companies between the process models that are supposed to be run and the one that actually runs [8]. This discrepancy is because some necessary tasks, internal control flows, and dependencies may not be included in the business processes models defined in the design phase. Therefore, if the future services are identified based on the decomposition of the predefined processes and not the ones that really run in systems, the identified services may not meet the business requirements. Besides, some tasks may never be considered as a potential service and are ignored.

Second, if the company's business requirements are changed, some parts of the business process may be affected and need to be altered. Therefore, having deep knowledge and understanding of the ongoing processes seems necessary to facilitate the changing process [9]. However, most companies do not have such sufficient knowledge about running processes or, if they have, it is not documented sufficiently or at least not presented explicitly. Hence, the changing process would be complicated, and following that may lead to some problems in finding new suitable services.

Third, suppose services are only identified based on the decomposition of process models without paying enough attention to other aspects of the business domain (e.g., business goals and data). In that case, the final identified services may not have enough precision and not be applicable.

Finally, the conceptual relations are the other subject that has not been considered sufficiently compared to structural ones in the strategy. Paying attention to such connections between tasks, goals, and data models and having data-aware business processes can lead to finding or defining new services [10]-[11].

All above mentioned challenges can effect on accuracy of the Method.

One solution to tackle these issues is combining process mining concepts and business process-driven strategy as a service identification method. In this regard, some open questions need to be addressed [9]. One of these questions is how to bundle the activities discovered from process mining as a potential service. The other one is recognizing reusable services that can be used out of context [9]. This study presents a new semi-automated service identification method by combining process mining techniques, goal model, and data model to answer these questions and the previously mentioned challenges. To this end, we choose data and goal models because they play a vital role in every process such that other aspects of a business domain would be meaningless without considering them. The proposed method includes three following phases:

In the first phase, the running business process and related knowledge are extracted from the information systems' event log. This phase, in turn, has three steps: (1) identifying workflow (without data), which is modeled by Petri-net, (2) validating the model by conformance checking techniques, and (3) enriching the model with finding related data to obtain a Business Process Model and Notation (BPMN) with the data. These steps are performed using process mining techniques and algorithms. These techniques help organizations to have a comprehensive view of the current business processes and associated activities [12]. Also, they provide various monitoring and improvement analysis tools. Furthermore, they help companies reengineer their processes to gain better performance, identify potential activities to be a service, change or reuse them with reasonable cost, and have the processes that address their business needs.

Phase 2 is dedicated to creating the goal and data model that helps extract the relations between tasks and identifying reusable services.

Finally, in the third phase, the mentioned models are linked by introducing different matrices. These matrices play a pivotal role in answering the mentioned challenges and service identification. Also, they help recognize semantic and structural relationships between tasks to identify other potential services. In the following, services are identified by presenting two algorithms. These algorithms use the mentioned matrices to find high cohesion and loose coupling tasks in their structural or conceptual dependencies, targeted common goals, or operating on mutual data. These tasks are assigned to different services based on their features. By following the above phases, the introduced setup questions are addressed. All the phases are covered in a case study, followed by listing the services and their assigned methods.

The remaining of this paper is structured as follows: Section 2 presents a quick background related to the research area. Section 3 describes details of our proposed approach through a case study. Finally, Section 4 is dedicated to the evaluation and conclusion. In the evaluation section, our method is compared with state-of-the-art service identification baselines based on the criteria introduced by some previous studies [7], [13], [14].

Related Work

Several service identification methods with a different strategy, input, or output have been presented in recent years [12]-[14]. Because of the diversity and multiplicity of these methods, our paper has limitations to inspect all of them. Therefore, only well-known methods that use business process identification strategies are discussed in this section.

Wang et al. [15] introduced an approach for service identification by making rules to design relations between business activities and potential services. They also designed ports, messages, and interfaces for identified services by considering both business and technical aspects. They presented a new algorithm that decomposes the BPMN model. However, their approach is not automated and has not been validated by a case study. Also, it has ignored other aspects of task dependency, such as the conceptual one.

Inaganti et al. [16] used some guidelines and technology to identify services by decomposing strategy. However, they did not introduce their measurement for recognizing potential business activities. Besides, there is no validating case study in their work. Indeed, their research work focuses on bid business-level services.

Jamshidi et al. [17] presented a cluster-based method that combines business entity strategies with business process decomposition. After modeling the business process, they identified and categorized the service model elements. The method was evaluated based on its use, users' analysis, and comparing with other existing methods. It is noteworthy that automation is not considered in this approach.

Dwivedi et al. [18] introduced a method that uses a heuristics algorithm for service identification. These researchers used the UML diagram as a business process model and applied it in a real example and model-driven development. Detailed information about the heuristics is not provided in this paper.

Bianchini et al. [19] used BP decomposition and ontology to recognize potential services and investigated process annotation semantically. Also, these researchers considered cohesion and coupling to determine services. Finally, they validated their method using a case study without considering automation.

BPA Onto SOA is the name of a method introduced by Yousef et al. [20]. Business process analysis and considering functional and non-functional business needs are the basis of this method. Also, Ontology, BP decomposition, and clustering techniques are applied to identify services; however, automation is missing in this method.

Azevedo et al. [21] chose candidate services using semantic point of view and heuristics algorithms. But detailed information and automation were not enough taken to account.

In [22], a clustering-based method is introduced. It considers services as many activities with high cohesion and loose coupling in their functionality. In this paper, BP decomposition and clustering techniques and algorithms are used. However, it does not consider other aspects of the business domain.

In [23], a method was proposed based on clustering and BP decomposition.

Moreover, in [20], 2PSIM was presented in which partitioning algorithms are applied to the BP model to identify services [24].

Kazemi et al. [25] introduced a method in the scope of automatic service identification using decomposition of business processes. But, it does not consider a different relation between tasks such as conceptual ones. In [26], the authors exhibited an automated model-driven service identification approach. This method uses the business model as an input and, after running heuristic algorithms, gives the service model as an output. AMSI is another automatic model-driven service identification method that identifies services by applying a multi-objective evolutionary algorithm. A high-level BP model is the input of this method [27].

El Amine et al. presented a method for service identification that uses both BP decomposition strategy and particle swarm optimization (PSO) algorithm [28]. A hybrid PSO algorithm was also applied in [29] for service identification. This algorithm addresses service design principles such as reusability, granularity, high cohesion, and loose coupling.

Alwis et al. [30] presented a heuristic algorithm for service identification based on business objects and their relationships. This method is considered a semi-automated approach that employs business processes. However, they did not assess the accuracy of their method, cohesion, and coupling metrics.

These authors also proposed another semi-automated heuristic algorithm that splits tasks into different categories based on their service identification functionality. This approach has a good performance for very large enterprises.

Nevertheless, due to the complexity of the method, it is not successful in addressing the needs of small to medium enterprises (SMEs) [31].

In [32], the authors provided a customized heuristic algorithm for service identification based on analyzing the business process codes. To this end, they developed a tool that converts a business model to a service model. However, this approach does not pay sufficient attention to service measurements like cohesion and coupling.

In [33], the authors provided an automatic tool that maps the BPMN component to software code for service identification. These methods only consider structural relations between tasks and ignore conceptual ones.

In [34], the authors proposed a framework that decomposes process and related logs with clustering algorithms. Then, an expert should modify the recognized services based on service features. This approach primarily depends on the process code and experts' opinions.

In [35], Leopold et al. introduced a new automated identification method that extracts a list of service candidates, including microservice, composite, and in hierarchy services. This method did not provide any solution for identifying the relationship between tasks.

Taibi et al. [36] proposed a method for service identification using business process flow extraction and clustering. After finding business process flow, they applied clustering to find the services. This approach does not consider system goals and data to find relations between tasks.

Lshob et al. [37] proposed a framework that produces a SOA model from the business process model. This approach employs some interfaces to convert a business process-based information system to an SOA-based one. Business requirements and the relation between activities are the missing aspects of this framework.

All the above methods employ a business process decomposition strategy to identify services. These process models are supposed to be run in the organization. But, in the run time and dynamic environments, most of them do not accurately and sufficiently address business needs [38]-[39]. Also, they may not show all tasks and related semantic and structural relationships that run in running time. Besides, since most of these methods do not have automated solutions to identify services, they are often descriptive and not sufficiently accurate.

Our literature review reveals that using running processes and taking other business processes into account can be a powerful technique for a more accurate service identification. However, the studies conducted in this regard are still scarce.

Proposed Approach

In this section, the proposed method is presented in three phases.

In the first phase, the ongoing business process model is extracted from the event log using process mining techniques and tools. After validating this model, it is applied as one of the inputs for the service identification phase.

In the second phase, the goal and data model are defined. These models help identify business and entity services. Moreover, using them allows finding conceptual relations and potentially reusable services.

Finally, in the third phase, services are identified by introducing some matrices and proposed algorithms. These matrices make a connection between BP, goal, and data model. Also, structure and semantic relations between tasks are identified using them. In this phase, the tasks that address the common goals or use the mutual data are considered the potential services method because of their internal conceptual relations.

Having real data-aware business processes, considering conceptual and structural interconnection between tasks (with the help of goal and data models and presenting semi-automated methods for service identification), leads to more accurate services.

Phase 1: Process Mining

Discovering an accurate business process model by process mining techniques needs different activities. They include collecting event logs and converting them to an acceptable format for process mining tools. Besides, applying discovery techniques and validating the output model is necessary in this regard [8]. In this phase, we investigate the mentioned activities.

In general, event logs generated by ERP systems, workflow management systems, or other information systems can be used as a primary input for process mining [8]. However, according to [40], they should be converted to the most common mining formats, i.e., extensible Markup Language (MXML) and extensible Event Stream (XES). There are different tools and methods for this purpose [8], [9], [40].

Since this paper focuses on service identification methods with the help of process mining techniques, we adopt a valid real-life event log format that is ready to use and exists in the process mining website[1]. This event log is related to the fine management process. The system's duty is to support and manage driving fines.

It contains activities, timestamps, and data related to the activities. We use the Prom tools for processing because it supports different process mining algorithms and related plug-ins [41].

[1] www.data.4tu.nl

Prom is one of the most popular and comprehensive tools that researchers use. The Manual and tool base/automatic part of this phase is shown in Table 1.

Table 1: Manual vs. automated part of Phase 1

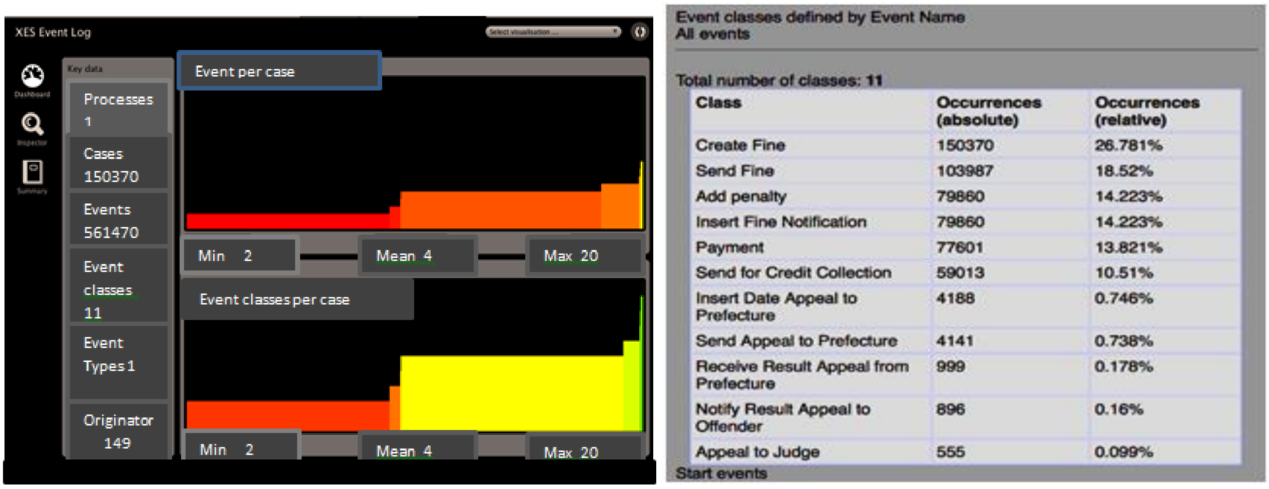
steps	User(manual)	Automated/tool support
1	Import the log to Prom for process	Processes the log to find activities and data in the output
2	Import the step 1 output to the mining algorithm	Processes the inputs, create process flow
3	Import both previous outputs to conformance checking algorithm	Processes both inputs, inspects and assesses the validation of the model
4	Applying data-aware algorithm	Generates process model with assigned data

Step 1: Finding Log Information

In this step, we inspect the log to discover activities and their assigned data using Prom tools. The results show that the event log contains 150,370 events. There are at least two and at most 20 trails of events in each set of events (called case). Also, the log demonstrates 11 different activities that happen in one process. These activities are created fine, send fine, add a penalty, fine notification, payment, credit collection, appeal to the prefecture, appeal to judge, send and receive appeal and notify result. Fig. 1 shows the extracted information about the log.

Step 2: Mining Process

In this step, we try to find the process flow in the event log by applying the inductive miner algorithm, which is one of the best techniques for finding business processes [42], [43]. There are different mining algorithms in the Prom. However, studies show that the Inductive miner produces more accurate and comprehensive results than others in dealing with noises and making models [42], [44]. The output is presented as a Petri net shown in Fig. 2.



Key data

Processes 1

Cases 150370

Events 561470

Event classes 11

Event Types 1

Originator 149

Event per case

Min 2 Mean 4 Max 20

Event classes per case

Min 2 Mean 4 Max 20

Event classes defined by Event Name
All events

Total number of classes: 11

Class	Occurrences (absolute)	Occurrences (relative)
Create Fine	150370	26.781%
Send Fine	103987	18.52%
Add penalty	79860	14.223%
Insert Fine Notification	79860	14.223%
Payment	77601	13.821%
Send for Credit Collection	59013	10.51%
Insert Date Appeal to Prefecture	4188	0.746%
Send Appeal to Prefecture	4141	0.738%
Receive Result Appeal from Prefecture	999	0.178%
Notify Result Appeal to Offender	896	0.16%
Appeal to Judge	555	0.099%

Start events

Fig. 1: Log inspection.

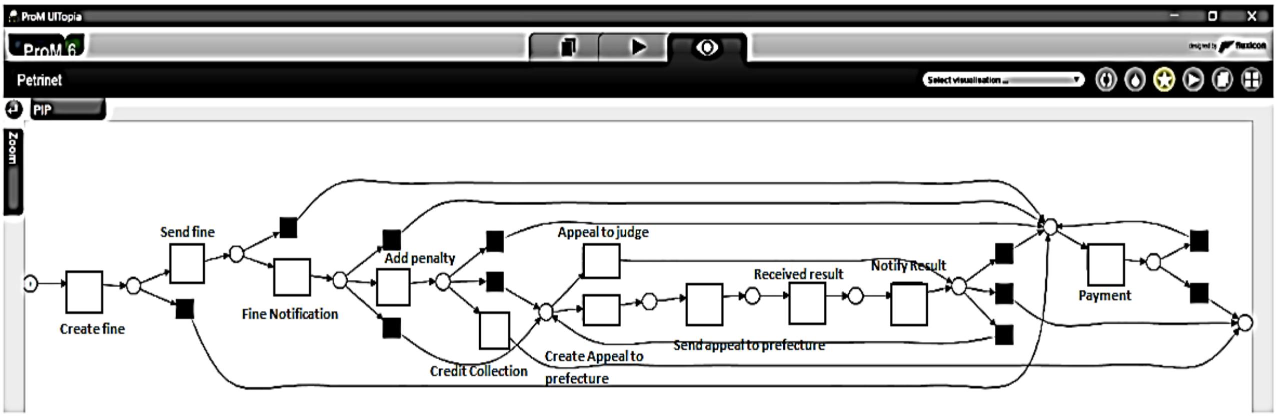


Fig. 2: The workflow created by the inductive miner algorithm.

Some black squares are placed parallel to other events in the figure, meaning that this event can be avoided, and another route can be taken. However, this rule does not apply to 'create a fine' and 'add the penalty' events. The output model needs to be checked for complying with the event log. This procedure, called conformance checking, can be performed using different algorithms and tools [45]. We use the "reply a log on Petri-net for conformance analysis" algorithms and plug-in, which is simple and the most accurate algorithms in Prom tools for this. Purpose [46] Conformance checking tries to answer the question of "what is the probability of re-establishing this process with this event log?". Fig. 3 presents a 91% fitness for this process, suggesting the validity of the extracted model.

Step 3: Data-Aware Process Mining

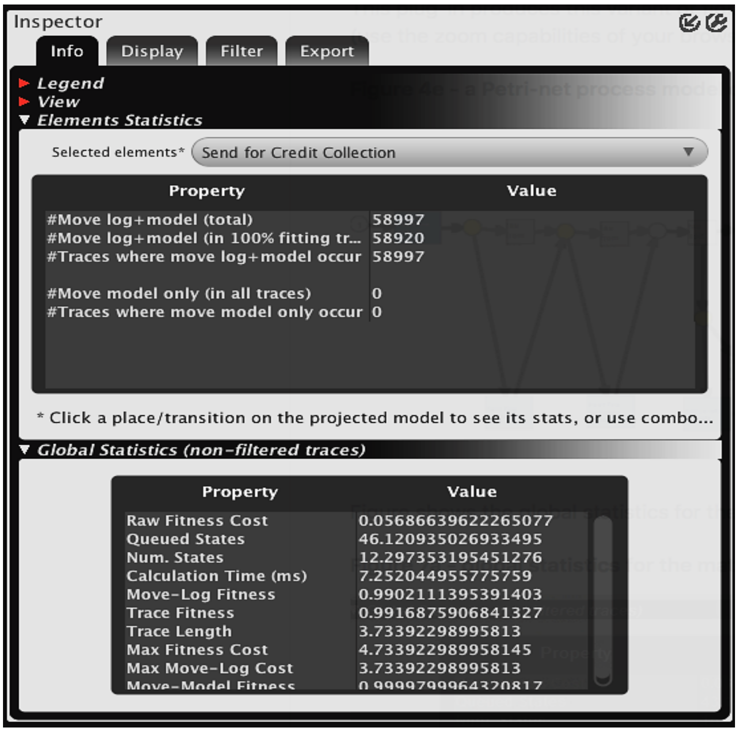
In this step, the log and the validated primary Petri net are imported to the Prom for detecting data. The output Data Petri net is shown in Fig. 4. In the resulting Data Petri net, each task is presented by a transition. Variables demonstrate the data associated with each transition. Here, transitions perform CRUD operations on data variables and identified data present attributes of the four main entities which flow in the process. For the sake of simplicity, this paper considers the entity instead of the attributes. The entities and the corresponding data models are shown in Phase 2 (Step 2).

Step 4: Mining BPMN

With the previous step's output and utilizing converting data Petri net to BPMN plug-in in Prom tool, the final BPMN model with data is mined. To check the validity of the outcome, we use BPMN to Petri net converter plug-in in Prom to see whether it generates the same data Petri net from obtained BPMN or not. The results analysis reveals that the BP model completely represents the same data as Petri net. If the model needs to be re-engineered for any reason (e.g., having better performance), it can be done in this step. For transforming As-is to the To-Be model, following the best practices proposed in [47] will be helpful. Following these rules, we create To-Be Models and the final output is demonstrated in Fig. 5.

Many plug-ins and algorithms in Prom directly produce the BPMN model from the event log [44], [48]. However, their output just shows workflows without presenting data, data streams, and their effect on decision gates. Having such a process model does not address the questions of this paper.

In this respect, by following our solution to reach the data-aware BPMN model, not only the accuracy and validation of the model in each step are checked but also data entities play a key role in discovering a more accurate BPMN model. Moreover, the Create, Read, Update and Delete operation (CRUD) on each entity is recognizable, which will be used in the service identification phase to find conceptual relations between tasks and entity services. It will be done by defining the entity-task relation matrix.



Selected elements* Send for Credit Collection

Property	Value
#Move log+model (total)	58997
#Move log+model (in 100% fitting tr...	58920
#Traces where move log+model occur	58997
#Move model only (in all traces)	0
#Traces where move model only occur	0

* Click a place/transition on the projected model to see its stats, or use combo...

Property	Value
Raw Fitness Cost	0.05686639622265077
Queued States	46.120935026933495
Num. States	12.297353195451276
Calculation Time (ms)	7.252044955775759
Move-Log Fitness	0.9902111395391403
Trace Fitness	0.9916875906841327
Trace Length	3.73392298995813
Max Fitness Cost	4.733922989958145
Max Move-Log Cost	3.73392298995813
Move-Model Fitness	0.9999799964320817

Fig. 3: Conformance checking by Prom.

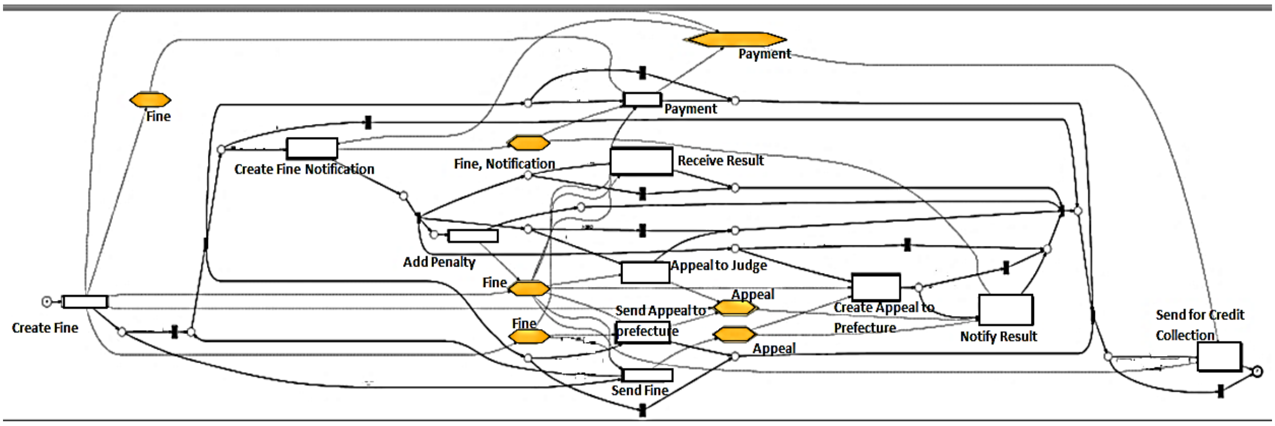


Fig. 4: Petri net with data.

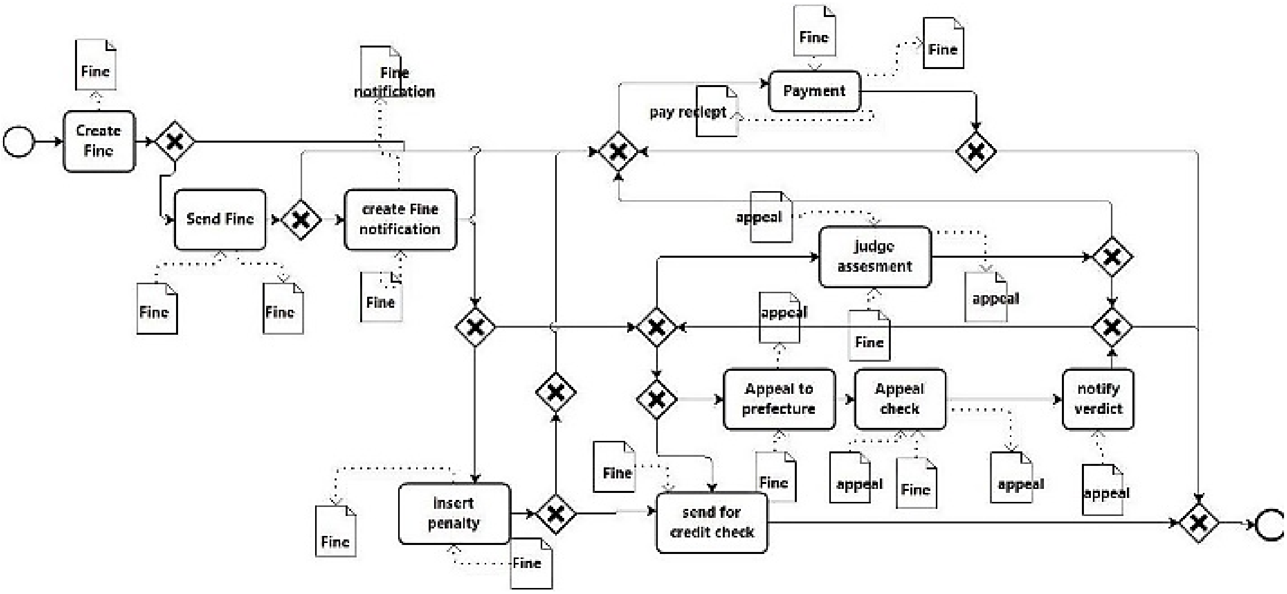


Fig. 5: The Business process model after applying process mining.

Phase 2: Modeling Goal and Data

This phase contains two steps: creating a goal and a data model. Table 2 presents the automated vs. manual parts of this phase.

Table 2: Manual vs. Automate parts of Phase 2

steps	Manual (user)	Automate
1	Create Goal Model	
2	Assess the output of the previous phase to create a data model	

Step 1: The Goal Model

In this phase, a goal model is created for the case study. Using this model helps find business requirements. Then, in the service identification phase, we will establish a relationship between the BPMN model's task and related requirements in the goal model by introducing the task-requirement matrix.

Such a relationship helps us to find the tasks with reusability features. The tasks addressing the same goals can be reused in similar service-oriented systems. There are different methods to creating the goal model [49]-[52]. We adopt GBRAM [50] in this research regarding its simplicity. In this model, only one identifier is considered for each requirement, and other characteristics of each need are omitted. The goal model in this method has different levels. The general goals are located at the highest level. These goals are decomposed into smaller ones to extract the requirements.

It is necessary to mention that if the goal model is documented and exists in the organization, it can be used as a basic model in this step, and there is no need to be recreated.

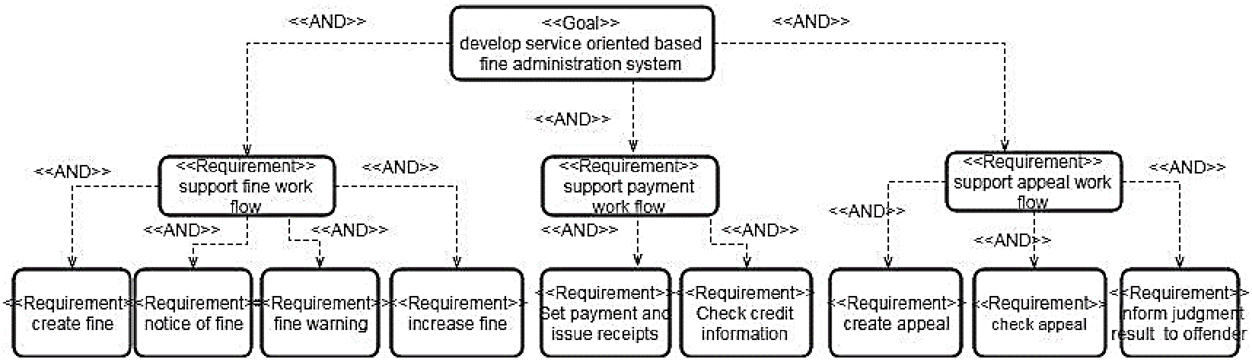

<<Goal>>
develop service oriented based fine administration system
<<AND>>
<<Requirement>> support fine work flow
<<Requirement>> support payment work flow
<<Requirement>> support appeal work flow
<<Requirement>> create fine
<<Requirement>> notice of fine
<<Requirement>> fine warning
<<Requirement>> increase fine
<<Requirement>> Set payment and issue receipts
<<Requirement>> Check credit information
<<Requirement>> create appeal
<<Requirement>> check appeal
<<Requirement>> inform judgment result to offender

Fig. 6: Goal model.

Step 2: Data Model

In this step, we create the data model by utilizing data that flow in the system. As mentioned in the last step of Phase 1, this model helps find entity services. The data model for the case study is depicted in Fig. 7.

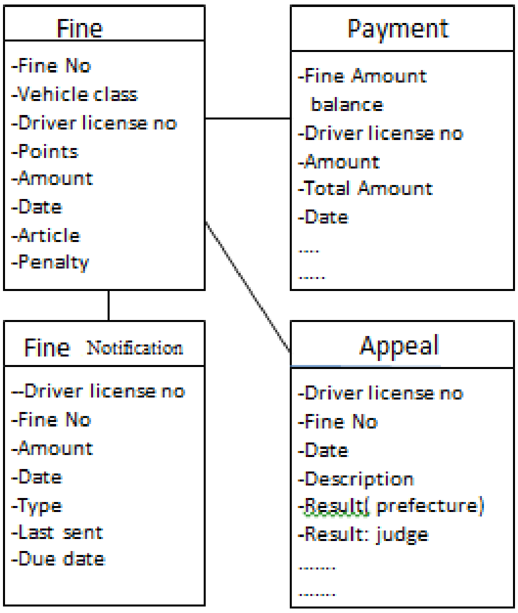


Fig. 7: Data model.

Phase 3: Service Identification

In this phase, services are identified based on data and goal models in the BPMN. Combining these models and using process mining techniques helps find both semantic and structural relations between tasks. In this paper, semantic relation is defined when either one or both of the following conditions occur: 1) two or more tasks either directly or indirectly (collaboratively) address the common goals and 2) when tasks do the CRUD operation on common entities. Considering these conceptual interrelations between tasks helps have highly coherent methods and more accurate services. Automated and manual parts of this phase are depicted in Table 3.

Table 3: Manual vs. Automated part of Phase 3

Steps	Manual (user)	Automate
1	Can be created by user or Tools (Visual Paradigm)	
2	Import required input (BPMN)	Process input, find relations Generate task-entity matrix Generate First relational matrix
3	Import requires input	Process input, find relations Generate a second relational matrix
4		Generate final relational matrix
5, 6	Import required inputs/Analyze the results	Finding services Merging services

Step 1: Create Requirements Task Array

In this step, a requirement- tasks two-dimensional array is created by adopting the goal model (Phase 2) and the BPMN model (Phase 1). This array shows the relationship between each task and related business needs.

To fill out the cells of this array, for each requirement set existing in the last level of the goal model, we investigate the tasks on the BPMN model; if the tasks support the needs, tasks are written in the related cell the array.

Having this array helps find reusable tasks. If organizations have the same goals, the tasks that address these goals can be reusable.

Therefore, identified services with this attitude meet the reusability factor, which is one of the service design principles. We use this array later to identify candidate services.

This matrix is indicated in Table 4.

Table 4: Requirements task array

Requirements	Create fine	Notice of fine	Fine warning	Increase fine	Set payment	Check credit	Create appeal	Check appeal	Inform verdict to the offender
Tasks	t1	t3	t4	t5	t2	t10	t6, t9	t7	t8

Step 2: Create First Task-Task Matrix

This step is divided into two-part. First, the relationship between tasks and related data in the BPMN model is investigated. For this purpose, we explore the data and associated tasks in the BPMN model and determine the type of operation that each task does on the related entities. As mentioned before, each task can perform different CRUD operations on entities. Then, an entity-task matrix is created. The matrix rows are tasks, and the columns are entities.

The cells show each task's type of operation on the related entities.

To quantify the relationship rate between tasks and the relevant entities, we need to convert their degree of relativity to a numerical value. For this purpose, we adopt this concept from [26]. Accordingly, it is assumed that CRUD operations have different degrees of importance. Their strength order is as C>U>D>R and their values (between 0 and 1) are determined as Create(c)=1, Update (U)=0.75, Delete= (D)=0.5 and Read(R)=0.25. As shown, the highest degree of importance is attributed to Create, and the lowest to Read [26]. The entity-task matrix for our case study is presented in Table 5.

Table 5: The task entity matrix

E1: Fine, E2: payment, E3: Notification, E4: Appeal

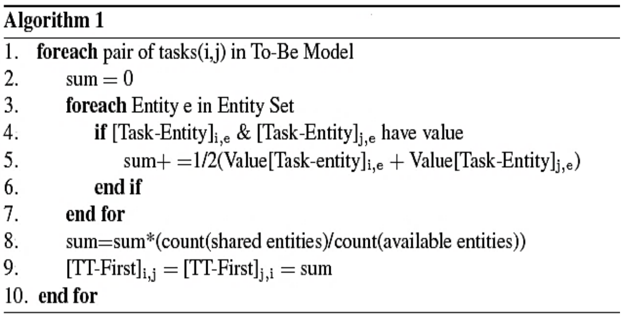
	E1	E2	E3	E4
t1	C			
t2	U	C		
t3	U			
t4	R		C	
t5	U			
t6	R			C
t7	R			R
t8				R
t9	R			U
t10	R			

In our case study, we consider each entity as a service to find entity services. Since each service comprises some methods, to determine them, in the task-entity matrix (Table 5), we look for the CRUD operations presented in the column associated with that entity. Entity services are derived from a business data model and can be reused to automate different business processes. Table 6 shows the entity services and related methods:

Table 6: Identified entity services

Methods	Entity services
Update (), Read (), Create ()	Fine
Create ()	Payment
Create ()	Notification
Create (), Read(), Update()	Appeal

In the second part, using the matrix shown in Table 5, the following algorithm created the first task-task matrix (Fig. 8).



```
Algorithm 1
foreach pair of tasks(i,j) in To-Be Model
    sum = 0
    foreach Entity e in Entity Set
        if $[\text{Task-Entity}]_{i,e}$ & $[\text{Task-Entity}]_{j,e}$ have value
            sum+ =1/2(Value$[\text{Task-entity}]_{i,e}$ + Value$[\text{Task-Entity}]_{j,e}$)
        end if
    end for
    sum=sum*(count(shared entities)/count(available entities))
    $[\text{TT-First}]_{i,j} = [\text{TT-First}]_{j,i} = \text{sum}$
end for
```

Fig. 8: The first task-task matrix algorithm.

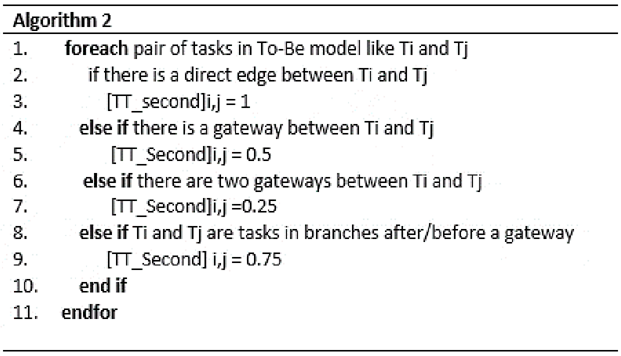
This matrix illustrates the relations between tasks based on their access to entities. Thus, it helps determine semantic relations between tasks. To develop this matrix, for every pair of tasks in the business process model, if they operate on one or some similar entities, the average of their accessing types to each common entity will be calculated. Afterward, these values will be added together. In the next step, the number of shared entities by these tasks is divided by the total number of entities accessed by these tasks. The result is multiplied by the value calculated in the previous step. This output is placed in the cell, i.e., at the intersection of the two tasks in the matrix. Applying this algorithm has two benefits. First, it helps understand tasks' relationships considering their impact on shared entities. Second, it allows finding the hidden and semantic relations between tasks. For example, if task x and task y impact similar data, they are conceptually related.

Applying the above algorithm and identifying such relations lead to identifying tasks that can be potentially considered for service detection (Table 7).

Step 3: Finding Structural Relation by second task-task matrix

In this step, the second task-task matrix is generated to illuminate the structural relations between tasks in the business process model.

Some analysis and design tools such as Visual Paradigm can automatically generate this matrix using the BPMN model; however, the matrix created by such tools has some shortcomings. For instance, it only considers the direct relationship between two tasks. Also, if a gate or event exists between them, they will not be considered as related tasks. Additionally, this tool neglects some series of patterns in business processes, such as tasks placed before or after a gateway that can be considered as services. We develop new tools that get the XML version of the BPMN model as an input to tackle these problems and analyze them. Then, it tries to find and quantify the structural relation between tasks by applying the following algorithm. In our proposed algorithm, if the connection between the two tasks of Ti and Tj is direct in the business process model, the value of the cell (Ti, Tj) in the second task-task matrix will be equal to 1. There should be a gateway between Ti and Tj, 0.5 will be assigned as the value. The value will be set to 0.25 if two gate ways are placed in a row and between the tasks. The value will be equal to 0.75 under the condition that Ti and Tj appear on branches positioned before or after a gateway (Fig. 9).



```
Algorithm 2
foreach pair of tasks in To-Be model like Ti and Tj
    if there is a direct edge between Ti and Tj
        [TT_second]i,j = 1
    else if there is a gateway between Ti and Tj
        [TT_Second]i,j = 0.5
    else if there are two gateways between Ti and Tj
        [TT_Second]i,j =0.25
    else if Ti and Tj are tasks in branches after/before a gateway
        [TT_Second] i,j = 0.75
    end if
endfor
```

Fig. 9: Quantify second task-task matrix procedure.

Table 7: First task-task matrix

	t1	t2	t3	t4	t5	t6	t7	t8	t9	t10
t1	0	0.437	0.875	0.312	0.875	0.312	0.312	0	0.312	0.625
t2	0.473	0	0.375	0.166	0.375	0.166	0.166	0	0.166	0.25
t3	0.875	0.375	0	0.25	0.75	0.25	0.25	0	0.25	0.5
t4	0.312	0.166	0.25	0	0.25	0.083	0.083	0	0.083	0.125
t5	0.875	0.375	0.75	0.25	0	0.25	0.25	0	0.25	0.5
t6	0.312	0.166	0.25	0.083	0.25	0	1.125	0.312	1.125	0.125
t7	0.312	0.166	0.25	0.083	0.25	1.125	0	0.825	0.75	0.25
t8	0	0	0	0	0	0.312	0.125	0	0.25	0
t9	0.312	0.166	0.25	0.083	0.25	1.125	0.75	0.25	0	0.125
t10	0.625	0.25	0.5	0.125	0.5	0.125	0.25	0	0.125	0

Table 8: Second task-task matrix

	t1	t2	t3	t4	t5	t6	t7	t8	t9	t10
t1	0	0.25	0.5	0	0	0	0	0	0	0
t2	0	0.25	0	0	0	0	0	0	0	0
t3	0	0.25	0	0.5	0	0	0	0	0	0
t4	0	0.25	0	0	0.5	0.25	0	0	0.25	0.25
t5	0	0.25	0	0	0	0.25	0	0	0.25	0.5
t6	0	0	0	0	0	0	1	0	0.75	0
t7	0	0	0	0	0	0	0	1	0	0
t8	0	0	0	0	0	0	0	0	0.25	0.25
t9	0	0.5	0	0	0	0.75	0	0	0	0.25
t10	0	0	0	0	0	0	0	0	0	0

Table 9: Final task-task matrix

	t1	t2	t3	t4	t5	t6	t7	t8	t9	t10
t1	0	0.723	1.375	0.312	0.875	0.312	0.312	0	0.312	0.625
t2	0.473	0.25	0.375	0.166	0.375	0.166	0.166	0	0.166	0.25
t3	0.875	0.625	0	0.75	0.75	0.25	0.25	0	0.25	0.5
t4	0.312	0.416	0.25	0	0.75	0.333	0.083	0	0.083	0.375
t5	0.875	0.625	0.75	0.25	0	0.5	0.25	0	0.5	1
t6	0.312	0.166	0.25	0.083	0.25	0	2.125	0.312	1.875	0.125
t7	0.312	0.166	0.25	0.083	0.25	1.125	0	1.125	0.75	0.25
t8	0	0	0	0	0	0.312	0.125	0	0.5	0.25
t9	0.132	0.666	0.25	0.083	0.25	1.875	0.75	0.25	0	0.375
t10	0.625	0.25	0.25	0.125	0.5	0.125	0.25	0	0.125	0

The output matrices generated by the suggested algorithms facilitate the identification of structural relations between tasks. Thus, the challenge of insufficient attention to the structural relationships between tasks in other service identification methods is overcome by creating this matrix. The second task-task matrix is presented in Table 8.

Step 4: Final task-task Matrices

The final matrix is equal to the sum of the first and the second communication task matrices, which shows the relationship between tasks from both structural and conceptual points of view. Therefore, each cell's value in the first and second communication matrix is summed and written in the corresponding cell in the final task-task matrix. Table 9.

Step 5: Recognizing candidate services

In this part, we create a service based on the business needs identified in Step 1 of Phase 1. For each business requirement, a new service is defined. Then, related methods for each service are determined by applying an algorithm presented in Fig. 10. According to this algorithm, the requirement-task array is first reviewed, and the tasks that address each business need are recognized and assigned to the related services. Suppose the algorithm faces some tasks previously assigned to another service. The first service with those tasks will be the owner of such methods in this situation. However, they are called in when their functions are needed to complete another service. Second, in this step, the algorithm reviews the final task-task matrix to find the remaining method for each service that is not assigned to any services.

Moreover, the algorithm looking for the tasks that may not be directly aligned with any business requirements, but are semantically related to the other methods, is identified for a particular service. This semantic relationship can be of two types: implicit control flow or data flow, both prerequisites for executing the identified service. Then, each task that remains unassigned to any service is allocated to a new service. Finally, the algorithm elucidates the relationships between the identified task services and other types of services such as entity and utility services. The candidate services for our example are shown in Table 10.

Table 10: Primary services for the case study

Methods	Services	
t1	Create fine	S1
t3	Notice of fine	S2
t5	Increase fine	S3
t4	Fine warning	S4
t2	Payment	S5
t10	Check credit	S6
t6	Appeal to prefecture	S7
t9	Appeal to judge	S8
t7	Appeal check	S9
t8	inform result	S10

After recognizing primary services, we investigate whether merging services is possible or not. To do so, we define variable RD as a ratio of dependency between services. It shows the ratio of the average internal correlation between each service's methods to the average connection degree between the methods of different services (1).

In some business processes, like our case study, only one task is suitable to consider as a method of the identified services. In such a circumstance, having fewer services with more abilities would be helpful to decrease the costs.

$$\mathrm{RD} = \text{average cohesion}/\text{average coupling} \qquad (1)$$

Variables of the above relation are defined in (2) and (3).

$$\text{Average Cohesion} = \frac{\sum \text{Cohesion}(S_i)}{N} \qquad (2)$$

where N is the total number of services; and

$$\text{Cohesion}(S) = \begin{cases} 1 & |S| = 1 \\ \sum\limits_{i,j} \text{task} - \text{task}(t_i, t_j)\ \forall t_i, t_j \ \in S & |S| > 1 \end{cases}$$

*If the service has one method, its cohesion is equal to 1. If the number of service methods is greater than 1, the formula calculates the value of cohesion.

and

$$\text{Average Coupling}(S) = \frac{\sum_{i,j} \text{Coupling}(S_i, S_j)}{D} \qquad (3)$$

where D is the total number of connections between services; and

$$\text{Coupling}(S1, S2) = \sum_{i,j} \text{task} - \text{task}(t_i, t_j)\ \forall\, t_i \in S1, t_j \ \in S2$$

Using above relations, we present a complementary algorithm (Fig. 11) to refine identified services based on these relations.

This algorithm needs a service dependency matrix as an input. The rows and columns of this matrix are the services. The matrix's main diagonal indicates the cohesion of methods within each service that can be calculated by cohesion(s) relation, which exists in (2). Other cells of this matrix are filled by the coupling variable's value, which is shown in relation (3). For our case study, this matrix is depicted in Table 11.

Table 11: Dependency matrix

	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	1	1.375	0.875	0.312	0.723	0.625	0.312	0.312	0.312	0
S2	0.875	1	0.75	0.75	0.625	0.5	0.25	0.25	0.25	0
S3	0.875	0.75	1	0.25	0.625	1	0.5	0.5	0.25	0
S4	0.312	0.25	0.75	1	0.416	0.375	0.333	0.083	0.083	0
S5	0.437	0.375	0.375	0.166	1	0.25	0.166	0.166	0.166	0
S6	0.625	0.5	0.5	0.125	0.25	1	0.125	0.125	0.25	0
S7	0.312	0.25	0.25	0.083	0.166	0.125	1	1.875	2.125	0.312
S8	0.312	0.25	0.25	0.083	0.666	0.375	1.875	1	0.75	0.25
S9	0.312	0.25	0.25	0.083	0.166	0.25	1.125	0.75	1	1.125
S10	0	0	0	0	0	0.25	0.312	0.5	0.125	1

After creating the dependency matrix, in this step, the value of the RD variable is calculated for the services using relation (1). If this ratio increases. Two services will be merged, and the set of services will be updated. The other situation is depicted in the algorithm

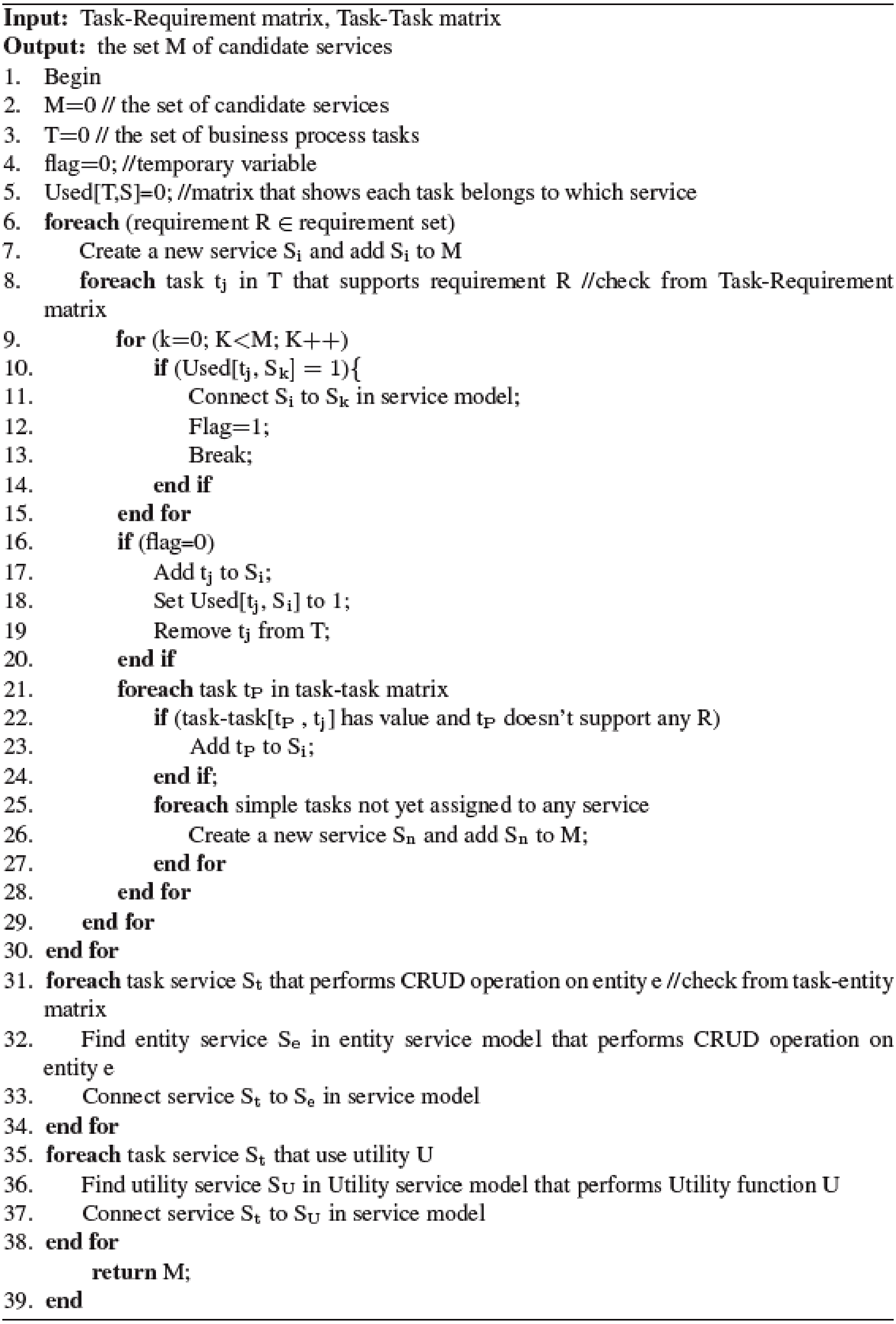
The final services for our case study are shown in Table 12.

Considering the definition of utility services from the identified services, each one can handle automatic tasks can be considered utility services.

For example, t3 can be considered a notification service, and when merged with t1, they can be identified as a composite service.

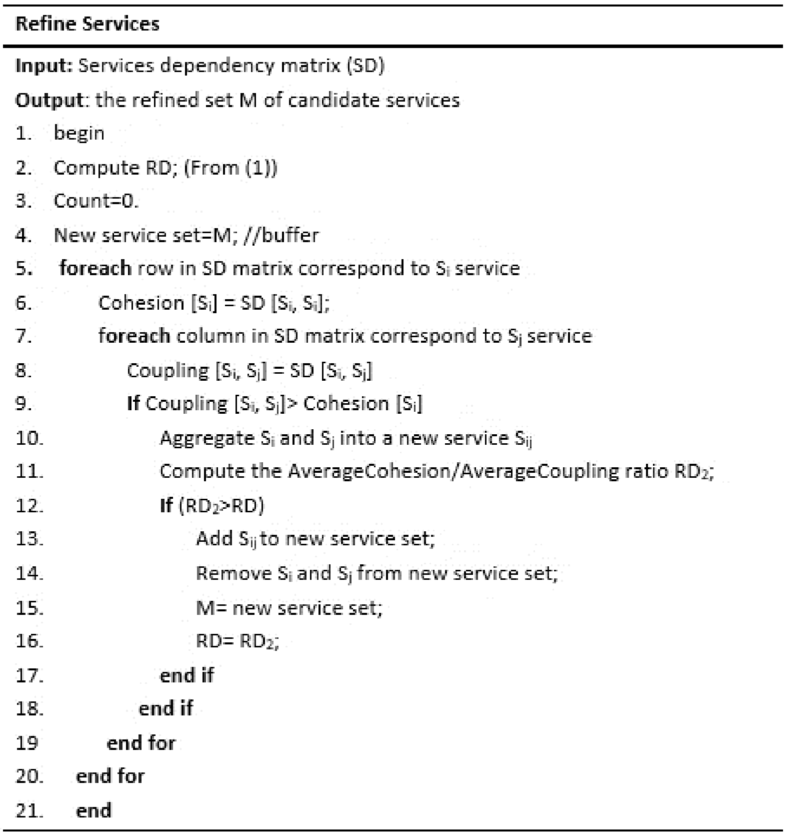
Table 12: Final identified services

services	Methods
S1	t1, t3
S2	t4
S3	t5
S4	t2
S5	t10
S6	t6, t7, t8, t9



```
Input: Task-Requirement matrix, Task-Task matrix
Output: the set M of candidate services
Begin
M=0 // the set of candidate services
T=0 // the set of business process tasks
flag=0; //temporary variable
Used[T,S]=0; //matrix that shows each task belongs to which service
foreach (requirement R $\in$ requirement set)
    Create a new service $S_i$ and add $S_i$ to M
    foreach task $t_j$ in T that supports requirement R //check from Task-Requirement matrix
        for (k=0; K<M; K++)
            if (Used[$t_j$, $S_k$] = 1){
                Connect $S_i$ to $S_k$ in service model;
                Flag=1;
                Break;
            end if
        end for
        if (flag=0)
            Add $t_j$ to $S_i$;
            Set Used[$t_j$, $S_i$] to 1;
            Remove $t_j$ from T;
        end if
        foreach task $t_P$ in task-task matrix
            if (task-task[$t_P$ , $t_j$] has value and $t_P$ doesn't support any R)
                Add $t_P$ to $S_i$;
            end if;
            foreach simple tasks not yet assigned to any service
                Create a new service $S_n$ and add $S_n$ to M;
            end for
        end for
    end for
end for
foreach task service $S_t$ that performs CRUD operation on entity e //check from task-entity matrix
    Find entity service $S_e$ in entity service model that performs CRUD operation on entity e
    Connect service $S_t$ to $S_e$ in service model
end for
foreach task service $S_t$ that use utility U
    Find utility service $S_U$ in Utility service model that performs Utility function U
    Connect service $S_t$ to $S_U$ in service model
end for
    return M;
end
```

Fig. 10: Service identification algorithm.



```
Refine Services
Input: Services dependency matrix (SD)
Output: the refined set M of candidate services
begin
  Compute RD; (From (1))
  Count=0.
  New service set=M; //buffer
  foreach row in SD matrix correspond to S_i service
      Cohesion [S_i] = SD [S_i, S_i];
      foreach column in SD matrix correspond to S_j service
          Coupling [S_i, S_j] = SD [S_i, S_j]
          If Coupling [S_i, S_j]> Cohesion [S_i]
              Aggregate S_i and S_j into a new service S_ij
              Compute the AverageCohesion/AverageCoupling ratio RD_2;
              If (RD_2>RD)
                  Add S_ij to new service set;
                  Remove S_i and S_j from new service set;
                  M= new service set;
                  RD= RD_2;
              end if
          end if
      end for
  end for
end
```

Fig. 11: Aggregation algorithm.

Results and Discussion

To evaluate our method, we sought to reach four different data. First, we needed to assess the accuracy of identified services, i.e., the number of services identified correctly based on their correct assigned tasks to the total number of services. Also, this kind of service should highly comply with business needs. Second, it was necessary to know how the complexity of the process can affect our method’s accuracy. Here, complexity included both process size and the number of gateways. Then, we needed to compare our proposed method with other service identification methods in case of common service quality metrics. Finally, we needed to examine the performance of the proposed method compared to other methods. In this paper, performance could be defined as the ability of the method to provide services at an acceptable level of accuracy and time. The ratio of accuracy to consumed time was calculated, and the results were expressed as a percentage in Table 14

In this study, ten different processes were given to five experts two times and asked to identify services based on their selected method. The processes had various degrees of complexity and sizes. On the first try, we gave them predefined business processes, and on the second try, we presented them with ongoing business processes resulting from process mining techniques.

Ultimately, we compare their results with the results arising from our proposed method. Average results are shown in Figs. 12, 13, and 14. As shown in the figures, the results of our method are more accurate than others (≈12%). and this superiority does not depend on the size or degree of complexity of the processes. Also, the results keep superiority when the basis process results from process mining techniques, and it shows about 8% improvement. In the next comparison, we compared our combination method with each goal, data, and business process-driven method separately as the basis of the proposed method. Next, we calculated the RD ratio for these service identification approaches on different processes. Simultaneously, we asked three experts to apply each approach to different given processes and rate them between 1 and 10 based on the quality of identified service and related assigned tasks. If they faced each misaligning task with a service identified by each method, they had to deduct one mark from their score. Finally, the average of scores was calculated for each approach. The result showed that our method's cohesion to coupling ratio was higher than others (≈7%). Moreover, results showed that the identified services were more compliant with the business needs whenever the base business process applied process mining techniques. Table 13 demonstrates the results.

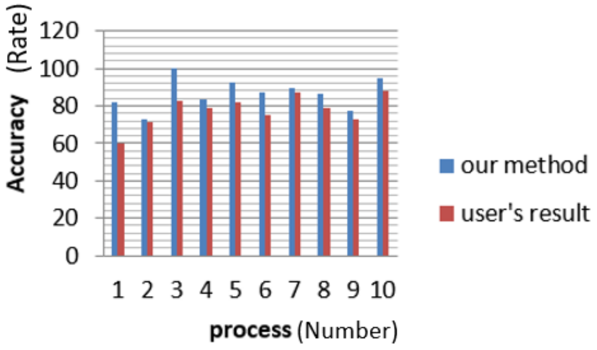


Fig. 12: Investigate the accuracy of the proposed method.

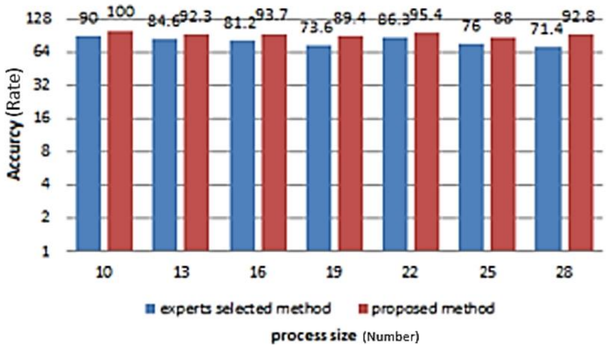


Fig. 13: Investigate accuracy and process size.

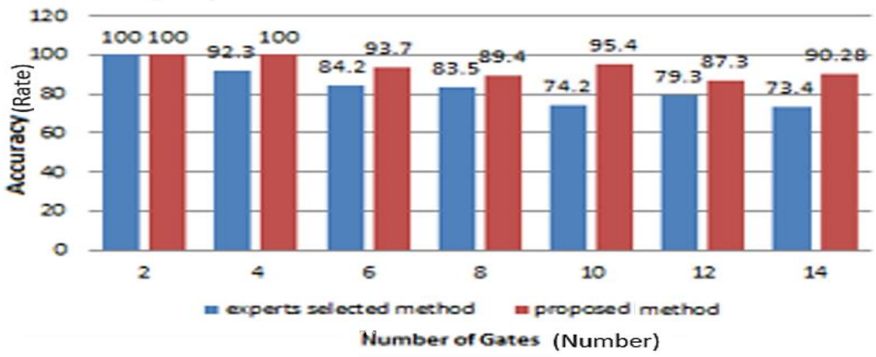


Fig. 14: Investigate accuracy and complexity.

Table 13: Comparing methods based on RD variable (PM: Process mining)

Method /Service	Business processes without PM	Goals	Entities	Business processes with PM	Proposed method
RD variable	2.146	1.092	1.726	2.274	2.389
Experts' evaluation	7.23	5.73	6.92	7.78	8.5

Table 14: Performance comparison between methods

Accuracy to time Ratio Average of Performance ≈ %	Business processes without PM	Goals	Entities	Business processes with PM	Proposed method
Process size <10	95.6%	92.24%	94.69%	97.47%	98.54%
Process size >10	77.25%	72.47%	76.33%	78.69%	79.96%
Number of gates <10	86.59%	83.59%	85.76%	88.36%	91.02%
Number of gates >10	76.59%	69.98%	74.49%	77.81%	82.67%

As mentioned earlier, in the Next step, the performance of our method has been calculated and compared with other methods. Table 14 shows that when the complexity of the process increases, the proposed method keeps the superiority over others to identify acceptable services in a more efficient time. In total, the average performance for the proposed method is almost 86% which is about 8% more than others.

In Table 15, our method is compared with other related methods from different perspectives, such as service guideline features, semantic and structural behavior, and automation.

As this table shows most of the service quality metrics have been considered in our method in comparison with others.

Table 15: Compare proposed method with others

	Strategy of identification	Identification technology	Type of service	Input	Standard s of model	Attention to structural behavior	Attention to semantic behavior	Attention to data structure	Level of automation	Cohesion	Coupling	Reusability	Granulity level
[21] Azevedo	Process Decomposition	Heuristic	Task, data, Composite	Process model	EPC FAD	✓	✓	✓	Non automative	-	-	✓	✓
[26] Jamshidi	Entities	Clustering Matrix	Software	CRUD process	-	✓	✓	✓	Automative	✓	✓	✓	✓
[53] Fareghzadeh	Process Decomposition, goal, Use-case, Legacy data	Analysis	Task, data, Composite, software	Use case, legacy system	UML	✓	✓	✓	Non-automative	✓	✓	✓	✓
[54] Kim	Goals and Scenario	Instruction	Software	Goal model	-	-	✓	-	Non-automative	✓	-	✓	-
[16] Inaganti	Process Decomposition	Instruction	Task	Process, Organization data model,	-	✓	-	✓	Non-automative	✓	-	✓	✓
[17] Jamshidi	Entities	Algorithm	Data, collaboration	Process	UML	✓	✓	✓	Semi-automative	✓	✓	✓	✓
[55] Chang	Process Decomposition	Analysis	Task	Process	UML	-	✓	-	Non-automative	-	-	✓	-
[56] Levi	Process Decomposition and Goals	Analysis	Task	Process, Goals	UML	✓	✓	-	Non-automative	-	✓	✓	✓
[57] Kazemi	Process Decomposition	Genetic algorithm	Task	Process	BPMN	✓	✓	-	Automative	✓	✓	-	✓
[58] Amiri	Process Decomposition	Heuristic-GA	Software	Process and goals	BPMN			✓	Semi-automative	✓		✓	✓
[30] Alwis	Process Decomposition	Heuristic	Task Composite Software	Process model	BPMN	✓	✓	✓	Non automative	-	-	✓	✓
[31] Alwis	Process functionality Decomposition	Clustering Heuristic	Software Task software	process	BPMN	✓			Automative			✓	✓
[32] Eric	Process Decomposition, Legacy code	Heuristic Algorithm	Composite, software	legacy system code	BPMN	✓	-	-	Semi-automative	-	-	✓	-
[33] Zafar	Process decomposition legacy code	Instruction	Software	legacy code	BPMN	✓	-	-	Semi-automative	-	-	✓	-
[34] Giseli	Process Decomposition	Clustering, Mining log	Task	Process,	BPMN	✓	-	✓	Non-automative	-	-	✓	✓
[35] Leopold	Process Decomposition	Algorithm	Task, software	Process, Legacy code	BPMN	✓	-	✓	automative		✓	✓	-
[36] Taibi	Process Decomposition	Clusteing, Mining process flow	Task, software	Process flow,	BPMN	✓	-	-	Semi-automated	-	-	✓	✓
[37] Leshob	Process Decomposition, legacy code	instruction	Task	Process, Legacy code	BPMN	✓	✓	-	automative	-	-	✓	✓
[59] Al Shereiqi	Process Decomposition	Clustering, Process mining	Task	Process	BPMN	✓			Non-automative	✓	✓	-	✓
Proposed method	Process Decomposition, Goals and entities	Algorithm, process mining	Task data utility, software	Goals, Process, Data	BPMN	✓	✓	✓	Semi-automative	✓	✓	✓	✓

Conclusion

This paper offers a new combined semi-automated method for service identification. The method considers different aspects of the business, including goals, data, and ongoing business processes, to recognize closely relevant tasks as a service. Since the Business process plays a key role in the method, we use process mining techniques to have deep and accurate knowledge about it. In the first phase, using such techniques, we extract the ongoing processes and associated data that flow in the organization from the real-life system event log. Then, we check the fitness of the output model with the verified conformance checking algorithms using Prom tools.

After validating the ongoing data-aware business process model, we create a goal and data model that helps find both structural and conceptual relations between tasks. These relationships' degree of correlation and coupling determines which tasks can be reused or recognized as services.

As a result, the process of moving to SOA will be easier, more flexible, and more accurate.

For this reason, the mentioned relations will be found by defining three matrices and considering different points of view: 1) addressing the same business needs (goal view and conceptual), 2) working on the same entity (data view and conceptual), and 3) structural interrelation between tasks in the business process model. Then, we link these matrices by introducing first, second, and final relational task-task matrices. These matrices’ cells show the dependency rate between tasks and use to understand cohesion and coupling relation between them by emphasizing conceptual and structural relations.

In the end, we propose two algorithms for identifying and merging candidate services that use the relational matrices mentioned earlier. These algorithms introduce a variable to measure the internal and external relationship between tasks. It means that if the cohesion value between tasks is more than the value of the coupling relation, the related tasks should be assigned to one service.

In this way, the cost of additional calls between services is decreased. The reusability feature (a crucial quality factor in service-oriented systems) is covered by tasks that address the same business needs. Whenever the business goals are the same in other processes, the related services can be replaced, or if the needs are changed, only its related services will be changed. Besides, working on common entities not only increases the cohesion between tasks but also helps increase the ability to reuse services.

To do all the subjects mentioned above, we received help from experts and different processes.

Our method is evaluated from different points of view. The results indicate the superiority of the proposed method in the case of accuracy, performance, and cohesion to cohesive ratio. Also, other comparing metrics states the same result.

Using the proposed method helps company to have more knowledge about the conceptual and structural relations between the tasks in the business processes. This capability enables them to reengineer these processes and improve them to have more accurate services.

As the method considers multiple aspects of the business domain, it can be extended.

Here, the potential path to extend this method for the future is suggested:

This method can be extended by considering other aspects of the business domain between tasks to reach high-quality services. Also, the method can be extended to cover other phases of service software development. In addition, the automated part of the method can be improved by decreasing human intervention; for example, by integrating the whole process through developing a tool that supports both the mining and identification phases.

Author Contributions

All the Authors contributed to all part of preparing and writing of this paper.

Acknowledgment

The authors would like to thank both the Editor and anonymous reviewers of JECEI for their valuable and constructive Comments and feedbacks.

Conflict of Interest

The authors declare no potential conflict of interest regarding the publication of this work. In addition, the ethical issues including plagiarism, informed consent, misconduct, data fabrication and, or falsification, double publication and, or submission, and redundancy have been completely witnessed by the authors.

Abbreviations

BPMN	Business Process Management
SOA	Service Oriented Architecture

References

[1] Y. Baghdadi, "A business model for B2B integration through Web services," in Proc. IEEE International e-Commerce Technology Conf.: 187-194, 2004.

[2] Y. Baghdadi, "Modelling business process with services: towards agile enterprises," Int. J. of Business Inf. Syst., 15(4): 410-433, 2014.

[3] Y. Baghdadi, W. Al-Bulushi, "A guidance process to modernize legacy applications for SOA," Serv. Oriented Comput. App., 9(1): 41-58, 2015.

[4] Q. Gu, P. Lago, "Service identification methods: a systematic literature review," in Proc. European Conference on a Service-Based Internet: 37-50, 2010.

[5] F. Kramer, C. Görling, S. Wind, "Service identification--An explorative evaluation of recent methods," in Proc. IEEE in 2014 47th Hawaii International Conference on System Sciences: 1285-1295, 2014.

[6] J. W. Hubbers, A. Ligthart, L. Terlouw, "Ten ways to identify services," The SOA Magz., (48), 2007.

[7] M. Abdellatif et al., "A taxonomy of service identification approaches for legacy software systems modernization," J. Syst. Soft., 173: 110868, 2021

[8] W. Van Der Aalst, "Process mining: Overview and opportunities," ACM Trans. Mng. Info. Syst. (TMIS), 3(2): 1-17, 2012.

[9] W. Van Der Aalst et al., "Process mining manifesto," in Proc. International Conference on Business Process Management: 169-194, 2011.

[10] M. Reichert, "Process and data: Two sides of the same coin?," in Proc. Springer OTM Confederated International Conferences, On the Move to Meaningful Internet Systems: 2-19, 2012.

[11] R. S. Huergo, P. F. Pires, F. C. Delicato, "A method to identify services using master data and artifact-centric modeling approach," in Proc. 29th Annual ACM Symp on Applied Computing: 1225-1230, 2014.

[12] W. Van Der Aalst, "Service mining: Using process mining to discover, check, and improve service behavior," IEEE Trans. Serv. Comput., 6(4): 525-535, 2012.

[13] B. Bani-Ismail, Y. Baghdadi, "A literature review on service identification challenges in service oriented architecture," in Proc. International Conference on Knowledge Management in Organization(Springer): 203-214, 2018.

[14] S. Cai, Y. Liu, X. Wang, "A survey of service identification strategies," in Proc. 2011 IEEE Asia-Pacific Services Computing Conference: 464-470, 2011.

[15] Z. Wang, X. Xu, D. Zhan, "Normal forms and normalized design method for business service," in Proc. IEEE International Conference on e-Business Engineering (ICEBE'05): 79-86, 2005.

[16] S. Inaganti, G. K. Behara, "Service identification: BPM and SOA handshake," BPTrends, 3: 1-12, 2007.

[17] P. Jamshidi, M. Sharifi, S. Mansour, "To establish enterprise service model from enterprise business model," in Proc. 2008 IEEE International Conference on Services Computing: (1): 93-100, 2008.

[18] V. Dwivedi, N. Kulkarni, "A model driven service identification approach for process centric systems," in Proc. 2008 IEEE Congress on Services Part II (services-2 2008): 65-72, 2008.

[19] D. Bianchini, C. Cappiello, V. De Antonellis, B. Pernici, "P2S: A methodology to enable inter-organizational process design through web services," in Proc. International Conference on Advanced Information Systems Engineering, Springer: 334-348, 2009.

[20] R. Yousef, M. Odeh, D. Coward, A. Sharieh, "BPAOntoSOA: A generic framework to derive software service oriented models from business process architectures," in Proc. Second International Conference on the Applications of Digital Information and Web Technologie (IEEE) : 50-55, 2009.

[21] L. G. Azevedo , "A method for service identification from business process models in a SOA approach," in Entrp. Business-Process and Info. Syst. Modeling. Spr.,: 99-112,2009

[22] Y. Kim, K. G. Doh, "Formal identification of right-grained services for service-oriented modeling," in Proc. International Conference on Web Information Systems Engineering (Springer): 261-273, 2009.

[23] M. Ren, Y. Wang, "Rule based business service identification using UML analysis," in Proc. 2nd IEEE Information Management and Engineering International Conf.,: 199-204, 2010.

[24] A. Nikravesh, F. Shams, S. Farokhi, A. Ghaffari, "2PSIM: two phase service identifying method," in Proc. OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"(Springer): 625-634, 2011.

[25] A. Kazemi, A. Rostampour, A. N. Azizkandi, H. Haghighi, F. Shams, "A metric suite for measuring service modularity," in Proc. CSI international symposium on Computer Science and Software Engineering (CSSE,IEEE): 95-102, 2011.

[26] P. Jamshidi, S. Mansour, K. Sedighiani, S. Jamshidi, F. Shams, "An automated service identification method," Technical report, TR-ASER-2012-01, Automated Software Engineering Research, TR-ASER-2012-01, 2012.

[27] M. Soltani, S. M. Benslimane, "From a high level business process model to service model artifacts-a model-driven approach," in Proc. ICEIS (3): 265-268, 2012.

[28] C. M. El Amine, S. M. Benslimane, "Using combinatorial particle swarm optimization to automatic service identification," in Proc. 13th International Arab Conference on Information Technology ACIT: 17-19, 2013.

[29] M. Mohamed, B. S. Mohamed, M. E. A. Chergui, "A hybrid particle swarm optimization for service identification from business process," in Proc. IEEE Second World Conference on Complex Systems (WCCS): 122-127, 2014.

[30] D. Alwis, A. Anuruddha, A. Barros, C. Fidge, A. Polyvyanyy, "Discovering microservices in enterprise systems using a business object containment heuristic," in OTM Confederated International Conferences, On the Move to Meaningful Internet Systems, Springer, 60-79, 2018.

[31] D. Alwis, A. Anuruddha, A. Barros, C. Fidge, A. Polyvyanyy, C. Fidge, "Function-splitting heuristics for discovery of microservices in enterprise systems," in Proc. International Conference on Service-Oriented Computing: 37-53, 2018.

[32] S. Eric, A. Moreira, C. Faveri, "An approach to align business and IT perspectives during the SOA services identification," in Proc. 2017 17th International Conference on Computational Science and Its Applications (ICCSA, IEEE): 1-7, 2017

[33] I. Zafar, F. Azam, M. W. Anwar, B. Maqbool, W. H. Butt, A. Nazir, “A novel framework to automatically generate executable web services from BPMN models,” IEEE Access, 7: 93653–93677, 2019.

[34] M. Gysel, L. Kölbener, W. Giersche, O. Zimmermann, "Service cutter: A systematic approach to service decomposition," in Proc. European Conference on Service-Oriented and Cloud Computing, Springer: 185-200, 2016.

[35] H. Leopold, P. Fabian, M. Jan, "Automatic service derivation from business process model repositories via semantic technology," J. Syst. Software, 108: 134-147, 2015.

[36] D. Taibi, K. Systä, " From monolithic systems to microservices: A decomposition framework based on process mining," in Proc. 8th International Conference on Cloud Computing and Services Science, 2019

[37] A. Leshob, R. Blal. H. Mili, P. Hadaya, O. Hussain "From BPMN models to SoaML models," in Proc. Conference on Complex, Intelligent, and Software Intensive Systems: 123-135,2019

[38] L. Jiang, J. Wang, N. Shah, H. Cai, C. Huang, R. Farmer, "A process-mining-based scenarios generation method for SOA application development," Serv. Oriented Comput. App., 10(3): 303-315, 2016.

[39] I. Zafar, F. Azam, M. W. Anwar, W. H. Butt, B. Maqbool, A. K. Nazir, “Business process models to Web services generation: A systematic literature review,” in Proc. IEEE, IEMCON. Conf . on Information Technology, Electronics and Mobile Communication: 789–794, 2018.

[40] O. AlShathry, "Process mining as a business process discovery technique," Compt. Eng. & Info. Tech., 5(1), 2016.

[41] B. F. Van Dongen, A. K. A. de Medeiros, H. Verbeek, A. Weijters, W. M. van Der Aalst, "The ProM framework: A new era in process mining tool support," in Proc. International conference on application and theory of petri nets: 444-454, 2005.

[42] S. J. Leemans, D. Fahland, W. M. Van Der Aalst, "Discovering block-structured process models from event logs-a constructive approach," in Proc. International conference on applications and theory of Petri nets and concurrency: 311-329, 2013.

[43] A. A. Kalenkova, W. M. Van Der Aalst, I. A. Lomazova, V. A. Rubin, "Process mining using BPMN: relating event logs and process models," Soft. Syst. Modeling, 16(4): 1019-1048, 2017.

[44] R. Ghawi, "Process discovery using inductive miner and decomposition," arXiv preprint arXiv:1610.07989, 2016.

[45] J. Carmona, B. van Dongen, A. Solti, M. Weidlich, Conformance Checking, Springer, 2018.

[46] W. Van Der Aalst, A. Adriansyah, B. van Dongen, "Replaying history on process models for conformance checking and performance analysis," Wiley Interdiscip. Rev.: Data Min. Knowl. Discovery, 2(2): 182-192, 2012.

[47] H. A. Reijers, S. L. Mansar, "Best practices in business process redesign: an overview and qualitative evaluation of successful redesign heuristics," Omega, 33(4): 283-306, 2005.

[48] A. A. Kalenkova, M. De Leoni, W. M. Van Der Aalst, "Discovering, analyzing and enhancing BPMN models using ProM," in BPM (Demos): 36, 2014.

[49] A. Van Lamsweerde, R. Darimont, E. Letier, "Managing conflicts in goal-driven requirements engineering," IEEE Trans. Softw. Eng., 24(11): 908-926, 1998.

[50] C. Rolland, C. Souveyet, C. B. Achour, "Guiding goal modeling using scenarios," IEEE Trans. software engineering, 24(12): 1055-1071, 1998.

[51] A. I. Anton, "Goal-based requirements analysis," in Proc. the second international requirements engineering Conf.: 136-144, 1996.

[52] H. Kaiya, H. Horai, M. Saeki, "AGORA: Attributed goal-oriented requirements analysis method," in Proc. IEEE joint international requirements engineering Conf.: 13-22, 2002.

[53] N. Fareghzadeh, "Service identification approach to SOA development," in Proc. World Academy of Science, Engineering and Technology, 35: 258-266, 2008.

[54] S. Kim, M. Kim, S. Park, "Service identification using goal and scenario in service oriented architecture," in Proc. IEEE APSEC'08. 15th Asia-Pacific Software Engineering Conf.: 419-426, 2008.

[55] S. H. Chang, S. D. Kim, "A service-oriented analysis and design approach to developing adaptable services," in Proc. IEEE Services Computing International Conf.: 204-211, 2007.

[56] K. Levi, A. Arsanjani, "A goal-driven approach to enterprise component identification and specification," Commu. ACM, 45(10): 45-52, 2002.

[57] A. Kazemi, A. Rostampour, P. Jamshidi, E. Nazemi, F. Shams, A. N. Azizkandi, "A genetic algorithm based approach to service identification," in Proc. IEEE (SERVICES), World Cong.: 339-346, 2011.

[58] M. J. Amiri, S. Parsa, A. M. Lajevardi, "Multifaceted service identification: process, requirement and data," Comp. Sci. Info. Syst., 13(2): 335-358, 2016.

[59] A. Al Shereiqi, Y. Baghdadi, "Business process mining for service oriented architecture," in ICT for an Inclusive World: (Springer): 3-19, 2020.

Biographies

Shahrzad Hekmat received her B.S. degree from Azad University, Iran, and her M.S. degree from Tehran University, in 2009 and 2011, respectively, both in Computer Engineering. She is currently studying her Ph.D. in the Department of Computer Engineering, Azad University Central Tehran Branch. Her research interests are in the areas of software engineering, service-oriented systems, and process mining.

- Email: shahrzad.hekmat@gmail.com
- ORCID: 0000-0002-3412-4576
- Web of Science Researcher ID: AEF-4134-2022
- Scopus Author ID: 57489672700
- Homepage: NA

Saeed Parsa received his B.Sc. in mathematics and computer science from Sharif University of Technology, Iran, his M.S. degree in computer science from the University of Salford in England, and his Ph.D. in computer science from the University of Salford, England. He is an associate professor of computer science at Iran University of Science and Technology. His research interests include software engineering, soft computing and algorithms.

- Email: parsa@iust.ac.ir
- ORCID: 0000-0003-4381-2773
- Web of Science Researcher ID: S-9536-2018
- Scopus Author ID: 8407441400
- Homepage: http://www.iust.ac.ir/page/717/Saeed-Parsa,-Ph.D.

Babak Vaziri received his B.S. degree in computer engineering from Shahid Beheshti University, Iran, and M.S. and Ph.D. degree in computer engineering from Islamic Azad University, Iran. He is currently an Assistant Professor of Computer Engineering in the Department of Computer Engineering, Central Tehran Branch, Islamic Azad University, Tehran, Iran. His research interests include software and process re-engineering and data mining.

- Email: vaziribabak@gmail.com
- ORCID: 0000-0002-8255-2794
- Web of Science Researcher ID: AGU-4445-2022
- Scopus Author ID: 57190976954
- Homepage: http://faculty.iauctb.ac.ir/b-vaziri-comp/fa

How to cite this paper:
S. Hekmat, S. Parsa, B. Vaziri, “A multi-aspect semi-automated service identification method,” J. Electr. Comput. Eng. Innovations, 11(1): 1-20, 2023.

DOI: 10.22061/JECEI.2022.8151.526

URL: https://jecei.sru.ac.ir/article_1703.html