Process mining-based business process management architecture: A case study in smart factories

Alireza Olyai[a], Shideh Saraeian[a*], Ali Nodehi[a]

[a]Department of Computer Engineering, Gorgan Branch, Islamic Azad University, Gorgan, Iran

salireza.olyaee@gmail.com, (+98)911-113-1473

shidehsaraeian@gmail.com, (+98)913-312-1630

alinodehi1396@gmail.com, (+98)911-371-5220

Abstract

Some business process management systems (BPMSs) have been developed in the field of smart factories. These systems are typically based on technical or production areas and technical processes. However, many existing systems, with respect to technologies used in smart factories and also the dynamic nature of the processes in these environments, are not able meet requirements of smart factories in the business process execution. The present study presents a new prototype of BPMS architecture based on smart factories' characteristics. This prototype has several components. In the monitoring component, process management can take place through process mining techniques inside a defined data analysis system for collecting event logs from big data. This component could operate based on control and optimization modules. The control module is applied to discover process models and their conformity with models extracted from business process analysis using Non-dominated Sorting Genetic Algorithm-II (NSGA-II) and Adaptive Boosting (AdaBoost) algorithms. Also, the optimization module can improve the processes model based on Business Process Intelligence (BPI) technique and Key Performance Indicators (KPIs). The results of the new prototype execution on a case study indicate that the proposed architecture is highly accurate, complete, and optimal in process management for smart factories.

Keywords: *BPMS; Dynamic business processes; Smart factory; Process mining; Big data*

1- Introduction

The Fourth Industrial Revolution (Industry 4.0) created the smart factory, including new technologies such as Cyber-Physical Systems (CPSs), the Internet of Things (IoTs), cloud computing, big data, and smart sensors. CPSs can monitor physical and production processes through different computer algorithms that communicate with each other and humans using the IoT structure. Also, produced organizational services could be applied with other partners using the cloud environment [1].

A Business Process (BP) contains a set of activities in an organizational environment to achieve a business goal [2]. Business Process Management (BPM) consists of principles, methods, and tools that integrate management sciences, Information Technology (IT), and industrial engineering [3]. In fact, BPM supports the life cycle of BP. In this case, BPMS can be used as a software tool for the technical support of BPM.

BPMS architecture [2,4] includes several components employed to hold up the BP life cycle. These components include a main component entitled process engine, which is applied for executing modeled BPs with a specific modeling language (e.g., Business Process Model & Notation 2.0 (BPMN 2.0)) [5]. Another key component is a user interface that interacts between users and the process engine for process management [6]. Also, control and monitoring of processes are other functions of inward BPMS. These processes can be done by data extraction from executed process instances. In this case, they have been considered to decide in the context of BPs optimal execution. These operations can be performed through a new group of techniques, such as process mining [7]. Process mining uses digital traces of processes to understand their visualizations and measure the performance of BPs [8]. The techniques like process mining have been applied as an appropriate tool in BPM to enrich the management of the smart factory in various aspects [9].

In this factory, process execution is in a smart status based on new technologies (e.g., big data, IoT, and cloud computing). In this case, big data analytics can provide useful information for information systems [10]. This function is possible by smartly collecting and analyzing huge amounts of data from various sources (e.g., market trends and future demands) [10]. Also, IoT, as a dynamic information network, includes objects connected to the Internet, such as sensors, actuators, and other smart equipment [11]. Besides, scalable resources have been provided using cloud computing dynamically [12]. Hence, due to the dynamic nature of smart factories' business and their various potentials, including controlling dynamic business processes [13], different systems and tools (e.g., BPMS) should be applied to control processes under dynamic conditions. Initially, a dynamic BP has not a precisely defined execution. Thus, its execution states are changed at runtime under new conditions (e.g., changes in business rules at runtime). In static business processes, however, the sequences of activities execution are determined at the early stage [14]. Many models and frameworks have been presented to describe smart factories' characteristics [15]. One the other hand, traditional approaches inward information systems architecture do not cover dynamicity in business process requirements [14]. As a result, classic information systems such as BPMSs do not cover these requirements adequately. Hence, the main gap is the absence of dynamic BPMS development in the smart factory.

To obtain competitive conditions in the market, rapid response to the customer, response to a wide range of changes in the factory environment, and further business-IT alignment, smart factories require a new generation of BPMSs to manage their BPs in dynamic conditions. Accordingly, this paper presents a new extension of a dynamic BPMS to BP monitoring and executing in a smart factory. Overall, the mentioned goal can be achieved by responding to the following research questions:

Rq1) What is the best architecture for designing a new extension of a BPMS based on smart factory characteristics?

Rq2) How can the use of the integration of big data analytics and process mining techniques in process monitoring inward the proposed BPMS appropriately?

The first research question considers the best architecture of BPMSs based on the smart factories' characteristics. In this case, new components are required to develop. The second research question focuses on the role of big data analytics and process mining techniques in processes monitoring inward proposed BPMS. The main contributions of this study are:

- Presenting a new architecture of BPMS which can apply in smart factories;
- Defining a data analysis system for collecting event logs from big data; and
- Monitoring the behavior of smart factories' processes through process mining techniques inside a defined data analysis system.

The remainder of this paper is organized as follows: Section 2 presents the literature review. Section 3 defines the proposed BPMS architecture. Section 4 provides the performance evaluations. Section 5 discusses the results. Finally, Section 6 presents the conclusions and future works.

2- Literature review

Several studies have been proposed by different researchers in the context of BPMSs, process mining techniques, and processes' optimization and control methods. In this section, previous BPMS-related studies are reviewed. These studies are described in the following:

2-1- Process mining techniques

Process mining is employed to analyze the behavior of an organization by extracting knowledge from event logs and using techniques to discover, monitor, and enrich process models [16]. Process mining consists of three sections: process discovery involves techniques employed to discover process models based on event logs information. Conformance checking includes algorithms used for reasons such as inspecting the conformance between the event log and the process model and checking the conformance between the discovered model and the observed behavior [17]. Enhancement contains techniques employed to enrich or develop process models.

Several process discovery algorithms, such as alpha, alpha+, heuristic mining, etc., have been presented in the literature. The alpha algorithm extracts process models by analyzing the relationships between activities in an event log. In this regard, Aalst et al. demonstrated the ability of this algorithm to discover a class of workflow processes [18]. One of the limitations of the alpha algorithm is that it does not detect short loops. To solve this problem, De Medeiros et al. presented an extended version of this algorithm called the alpha+ algorithm [19]. A heuristic mining algorithm [20] can process noise event logs. The disadvantage of this algorithm is the inability to identify non-free choice or non-local structural patterns.

One of the methods of discovering process models is using meta-heuristic algorithms. In this regard, de Medeiros et al. [21] applied the genetic algorithm to discover process models. They used the genetic algorithm to take advantage of the global searches performed by these types of algorithms. In [22], a combination of Particle Swarm Optimization (PSO) and simulated annealing algorithms was suggested for process mining and extraction of optimal process models. The researchers' goal in the proposed method was to improve the execution time of the algorithm and the quality of the extracted models. Also, Alizadeh and Norani [17] proposed a new algorithm called ICMA to discover process models. To this end, they integrated the imperialist competitive algorithm into their proposed algorithm. Xu et al. [23] proposed a combination of alpha and genetic algorithms to discover process models. They believe that the proposed algorithm provides better performance than the alpha algorithm.

Some conformance-checking algorithms have been proposed in the literature, including Footprint, Replay, and Alignment. In [7], these approaches are mentioned and discussed. Also, in [24], a new token-based replay technique is suggested to increase speed and scalability in the field of conformance checking. They believed that this approach could provide more accurate diagnoses, thereby preventing problems such as token flooding. This new technique is implemented by the PM4Py library. Burattin et al. [25] presented an online conformance-checking method using behavioral patterns evaluation. These authors aimed to identify deviations online so that they can be retrieved before the execution of the process instance is completed. De Leoni et al. [26] presented a method for aligning event logs and process models by considering perspectives such as data and resources. According to these authors, unlike other conformance-checking methods that focus on control flows, the proposed method also considers data and resources when conformance checking of process.

2-2- BPs monitoring methods

In the modern business world, organizations must reorganize BPs to achieve success in the market [27]. This goal is achieved by monitoring processes that include control and optimization. Saraeian et al. [28] developed a new controller component for inward uncertain BPMS. This component monitors the critical infrastructure in the automatic closed-loop supply chain. Vera-Baquero et al. [29] presented an architecture that integrates big data analytics and BPM in a distributed environment. In this case, users can analyze the results of BP execution. Also, different tools (e.g., SAS business intelligence and analytics) are presented to control the processes. Pourmirza et al. [30] designed a BPMS reference architecture that can control the behavior of process instances by using such tools. Krumeich et al. [31] proposed hybrid architecture of big data analysis methods and complex event processing techniques to control processes in a sample company in the field of Industry 4.0. In [32], a solution was proposed for re-engineering BPs and optimizing them. The proposed approach can by identifying the priority of the activities, detect the insignificant activities that consume a lot of time and resources. Duran et. al [33] presented a method for analyzing BPs based on a machine-learning algorithm.

In the context of meta-heuristic methods, Vergidis and Tiwari [34] used a developed version of the genetic algorithm, i.e., NSGA-II, to optimize the features of the BPs design. This optimization aims to design BP through optimal features such as cost and process execution time. In [35], a resource allocation method was developed to optimize resource allocation using the improved PSO algorithm. Also, the proposed method considers different indicators of process performance evaluation, such as resource cost and time. Mahammed et. al [36] presented an extended version of genetic algorithm to optimize BPs. This method uses a multi-population genetic algorithm for optimal design of processes.

As mentioned earlier, process mining techniques are among the methods of discovering and optimizing business processes [22]. Jiang et al. [37] presented a new method to analyze and optimize BP models using the process mining technique. This method is based on service-oriented architecture. Also, Yang et al. [38] proposed an architecture to analyze production processes through big data and process mining techniques. Hongtao et al. [39] defined a process mining architecture based on BPs optimization. This architecture uses knowledge about processes and hidden relations for optimization goals. This architecture employs a measurement module to investigate the optimized process's performance.

2-3- Architectures of BPMSs

2-3-1- Reference architectures

In terms of the reference architecture of BPMS, Hollingsworth provided workflow reference architecture for Workflow Management System (WFMS) entitled workflow management coalition. This researcher also introduced essential components and interfaces of the workflow management system [40].

As shown in Figure 1, this architecture consists of several components with different functionalities, such as analysis, modeling, and description of BP using process definition tools. Workflow API and interchange formats can regulate the relationship between system components and workflow control software. Also, workflow execution service includes one or more workflow engines employed to create, manage, and execute workflow instances. In addition, workflow client applications include software that interacts with the end user. Moreover, invoked applications include services, applications, or invoked programs for different purposes. Finally, the administration and monitoring tools component can monitor and manage workflow engines.

Grefen and de Vries presented Mercurius as another reference architecture. This reference architecture was designed for a mature WFMS for mobile workflow customers in heterogeneous environments [41]. BPMSs have been becoming pervasive since 2005. These systems are mostly based on service-oriented architecture (SOA) technologies and protocols. In this respect, Arsanjani et al. [42], proposed a reference architecture for these systems, entitled service-oriented solution stack. This system, provides reference architecture based on SOA in nine layers to enrich the business. A review of different BPMS architectures characteristics is presented in Table 1.

Although each of the above architectures considered significant points (e.g., flexibility, proper interaction with end-user, etc.), the critical points that are less addressed are smart factories' dynamic and scalable conditions. In real life, managing and monitoring smart factories' changes is essential. Therefore, considering dynamic conditions in architectural modeling is a research gap in this field.

2-3-2- Commercial BPMSs

Meidan et al. introduced varieties of open-source or commercial BPMSs that any organization can use [50]. A review of the features of some of these BPMSs is presented in Table 2.

Commercial BPMSs cannot support dynamic processes [51] and can primarily act in a static state. Also, due to the dynamicity of business rules in dynamic processes [50], business rule changes cannot be supported in commercial BPMS. Thus, these types of BPMS do not have optimal functionality in smart factories.

2-3-3- Academic BPMSs

Delgado et al. [6] created a generic BPMS user portal using the integration of the process engines (such as Bonita and Activiti). This BPMS consists of a process engine and a web user portal. Saraeian et al. [52] designed an uncertain BPMS based on a new engine through different standards and interfaces definitions. This BPMS was implemented for managing uncertainty at runtime. Schulte et al. [53] designed an elastic BPMS to manage processes that are executed using cloud resources. This architecture has different capabilities, such as scheduling and decentralized coordination. Alexopoulou et al. [54] proposed another BPMS architecture using the approach of events-based process modeling for dynamic processes. These models require BPMS architecture with an engine that acts based on receiving events. Vasilecas et al. [14] proposed a new simulation and modeling method of dynamic BPs based on context changes and business rules changes at runtime.

Based on the above studies, it is concluded that when an event happens, the BPMS must be able to record information related to the event and notify other sections to perform some functionalities to deal with it [54]. As a result, employing a classic BPMS to execute dynamic processes is inappropriate and avoids flexibility [54]. Hence, there is a need for a BPMS that fully supports the changes in business rules due to the dynamic conditions of smart factories.

2-3-4- BPMSs applications in smart factories

Researchers have presented different approaches to improve smart factories challenges through BPMSs. Kozma et al. [55] proposed a workflow based on SOA for production systems. They believe that the proposed approach can meet the requirements of smart factories such as decentralization, modularity, etc. Seiger et al. [56] presented an architecture integrating Industrial IoT (IIoT) and BPM for smart factories. This approach shows the benefits of employing BPM technology for production processes in IIoT. Gorski et al. [57] presented a workflow for predictive maintenance in support of manufacturing operations. The results show that this method can enrich maintenance processes. In [58], a workflow architecture is suggested for a production environment in smart factories. This architecture can provide benefits such as reducing the time and cost of transactions. Li et. al [59] proposed a new workflow for production scheduling in smart factories. This method can provide optimal scheduling for production in these environments. In this research, the authors have used Genetic and Tabu Search algorithms for optimization.Table 3 provides an overview of BPMSs application inward smart factories.

In general, this research aims to monitor BPs and present an architecture based on the dynamic and scalable structure of smart factories. In the field of BP monitoring, previous studies did not consider process control and optimization together. Therefore, in this study, control and optimization of processes are considered simultaneously in the monitoring component. Also, the proposed method fully supports improving BP rules integrated with big data analysis to obtain better accuracy and efficiency. Regarding architectural design, by studying commercial BPMSs, academic BPMSs, and reference architectures, it was concluded that the existing designs did not consider the dynamic and scalable conditions of smart factories as they should. Thus, the present research offers a new architecture to model the dynamic conditions of smart factories.

3- Proposed BPMS architecture

As mentioned before, to fill the mentioned gap, the present study develops a new prototype of BPMS architecture. This architecture aims to enrich the existing architectures to employ an architecture in smart factories to execute and control business processes. Figure 2 illustrates different defined components in the new architecture.

These components include several modules interacting with the architecture's central core. The process enactment service includes one or more process engines employed to execute BP instances. Also, dynamic BPs should react to business rules, and environment changes at runtime [14]. In this case, business rules and events of the business environment should be defined in the rules engine component. Accordingly, after defining business events, a set of rules are selected to execute the next activities in BP. The client applications component is another main component containing different applications used for user interactions with process engines to perform desired activities.

The present study developed new components (e.g., the modeling and monitoring components) containing several modules. They are described in more detail later.

3-1 Modeling Component

Smart factories encounter dynamic processes that can use real-time data generated with the IoT. Thus, BPMN 2.0, as a common modeling language, is unsuitable for our purpose due to insufficient elements for this type of modeling. Therefore, the present study developed the BPMN 2.0 language with accepted new elements of reference [64] that provides different elements for modeling this type of process. These elements are presented in Table 4.

Therefore, the extended version of the BPMN2.0 language could be modeled different dynamic processes inward smart factories.

Also, the metamodel of the new extended modeling language from the BPMN2.0 language is presented in Figure 3.

3-2 Data Analysis

Due to using new technologies such as the IoT and big data in smart factories, there is a large amount of data for mining. In addition to the structured data, streaming data related to IoT (such as IoT sensor data) are also considered in the proposed architecture. Since traditional approaches to batch data processing cannot be used for streaming data, big data analysis techniques can be used to process streaming data. Figure 4 demonstrates the method used to analyze big data.

As shown in Figure 4, the proposed data analysis method considers both structured data and streaming data. First, to analyze streaming data, parallel and real-time processing is performed on this data using Apache Spark Streaming tool. The result of the processing is stored in the Hadoop HDFS file system. Then, from the structured data, event logs are extracted and stored in HDFS. In this study, the Apache SparkSQL tool was used to create a query on HDFS. Using the above tools allowed for preparing log files for the process mining.

3-3 Monitoring Component

The proposed monitoring component in the new architecture provides the control of executed processes status. This component includes control and optimization modules with several functionalities, described in the following sections.

3-3-1 Control Module

In the proposed architecture, the control module allows controlling the status of executing processes. This control includes two types of functions: 1) a control of the behavior of executing process models and 2) a control of running process instances. First, the behavior of the process models is evaluated using the Process Model Controller function, and then the Process Instance Controller function controls the running process instances. Hence, these functions' inward control module provides strong support for monitoring processes.

3-3-1-1 Process model controller function

As mentioned earlier, the executable data is provided as an event log concerning BPMS. The process model controller function could be controlled the behavior of BPs through the process mining technique. Thus, conformance checking step inward process mining can compare process model execution with past process models extracted from event logs using the discover operation. The executions can continue if the process models run truly; otherwise, the improvement operation should be performed for investigated process models (Figure 5).

In this paper, NSGA-II, as an extended version of the Genetic Algorithm (GA), has been used to discover the optimal process models. The advantages of this algorithm are as follows:

- It provides a solution close to the Pareto optimal solution using non-dominated sorting techniques.
- It uses crowding distance techniques to maintain diversity in solutions.
- It preserves the best solution of the current population in the next generation by using elitist techniques.

The remarkable point is that conventional process mining algorithms can produce a single process model that may not describe recorded behavior well. In this case, NSGA-II could be employed for generating several process models from event logs [65]. In general, a process model can be defined as a casual matrix as follows [22,66]:

$$CM = \{(In(a_i), Out(a_i)), \forall i = 1...n\}$$

where a_i is considered an activity (or task) in the event log, $In(a_i)$ includes the set of activities preceding the a_i activity, $Out(a_i)$ includes the set of activities following the a_i activity, and n is the total number of activities. Also, in discovering process models using NSGA-II, individuals are casual matrixes. In other words, all individuals in a population are defined by a set of activities. The flowchart of this algorithm is presented in Figure 6.

In the process mining technique, NSGA-II starts by creating an initial population of individuals (chromosomes). Each individual is associated with a process model. This algorithm can calculate the fitness metrics for each individual. Also, the algorithm performs the fitness evaluation and sorting of the population based on a fitness function and dominating conditions, respectively. In this case, two individuals in the population who have the highest fitness value are selected as parents. This selection should be performed based on the lower individual`s rank and its high crowding distance.

In the next step, the initial population is integrated using an obtained population from the mutation and crossover operators. Thus, the members from the top of the sorted list are selected, and other remaining population members are discarded. These selected members can make the next generation of the population. All the mentioned steps are repeated until the desired generation (the most optimal individuals) as models can be employed.

Also, crossover and mutation operators were used to create the individuals of the next generation. Crossover is used to recombine existing individuals in the current population. In fact, these individuals can be generated by combining a subset of the causality relations in the population. The pseudo-code of the crossover operator is shown in Algorithm 1.

Algorithm 1. Crossover operator [21, 67]

Input: Two parents (e.g., $parent_1$ and $parent_2$), crossover rate

Output: Two recombined offspring (e.g., $offspring_1$ and $offspring_2$)

1. $Offspring_1 \longleftarrow parent_1$ and $offspring_2 \longleftarrow parent_2$.
2. With probability crossover rate:
   a. Select (Randomly) a task (e.g., a) to the crossover point of the offspring.
   b. Select (Randomly) a swap point sp_1 for In_1(a). (The swap point goes from position 1 to n-1, where n is the number subsets in the condition function In_1(a)).
   c. Select (Randomly) a swap point sp_2 for In_2(a).
   d. $remainingSet_1$(a) equals subsets in In_1(a) between positions 0 and sp_1.
   e. $swapSet_1$(a) equals subsets in In_1(a) whose position equals or bigger than sp_1.
   f. Repeat steps 2d and 2e but use $remainingSet_2$(a), In_2(a), sp_2 and $swapSet_2$(a) instead of $remainingSet_1$(a), In_1(a), sp_1 and $swapSet_1$(a).
   g. For every subset S_2 in $swapSet_2$(a), do:

      With equal probability, perform one of the following steps:
      A. Add S_2 as a new subset in $remainingSet_1$(a).
      B. Join S_2 with an existing subset X_1 in $remainingSet_1$(a).
      C. Select a subset X_1 in $remainingSet_1$(a), delete the elements of X_1 that are also in S_2, and add S_2 to $remainingSet_1$(a).
   h. Repeat step 2g but use S_1, $swapSet_1$(a), X_2 and $remainingSet_2$(a) instead of S_2 and $swapSet_2$(a), X_1 and $remainingSet_1$(a).
   i. $In_1(a) \longleftarrow remainingSet_1$(a) and $In_2(a) \longleftarrow remainingSet_2$(a).
   j. Repeat steps 2b to 2h but use Out(a) instead of In(a).
   k. Update the related task to a.
3. Return $offspring_1$ and $offspring_2$.

Also, in the mutation, some changes occur in an individual. In other words, the mutation operator may change a population's existing casualty relations. Algorithm 2 presents the pseudo-code of the mutation operator.

Algorithm 2. Mutation operator [21, 67]

Input: An individual mutation rate

Output: A mutated individual

1. For every task (e.g., a) in the individual, do:
   a. With probability mutation rate, do one of the following operations for the condition function In(a):
      i. Select a subset X in In(a) and add a task, e.g., a’ to X, where a’ belongs to the set of tasks in the individual.
      ii. Select a subset X in In(a) and delete a task, e.g., a’ from X, where a’ belongs to X. If X is empty after a’ removal, exclude X from In(a).
      iii. Distribute the elements in In(a) again.
   b. Repeat step 1a, but use the condition function out(a) instead of In(a).
   c. Update the related tasks to a.

The discovered process models should be applied in comparison with executing process models. This comparison was made through the AdaBoost algorithm. AdaBoost is one of the important algorithms in the machine learning field with significant characteristics such as computational accuracy and simplicity [66]. Algorithm 3 presents the AdaBoost algorithm to check the conformity of process models.

Algorithm 3. Pseudo-code of AdaBoost algorithm [66]

Input:
T (maximum number of iterations), n (number of activities), t = 1, and z (normalizer);
Train dataset ⟵ discovered process model;
Test dataset ⟵ executing process model;
Labels of samples ⟵ (a_i, In(a_i), Out(a_i));

Given a discovered process model {(a_i, (In(a_i), Out(a_i))), ∀ i = 1…n} as a training set;

Initialize the observation weights: $w_t(i) = \frac{1}{n}$;

For t = 1… T

1.Train a weak classifier such as $h_t(a)$ by weighted training data to minimize error;

2 .Calculate the voting weight of $h_t(a)$: $\alpha_t = \frac{1}{2} \log (\frac{(1-e_t)}{e_t})$;

3. Calculate the new weights of training samples to increase the misclassified sample weights and decrease the incorrect classified sample weights ($W_{t+1}(i) = w_t(i) \exp \{- \alpha_t \frac{(\mathrm{In}(a_i),\mathrm{Out}(a_i))h_t(a)}{z_t}\}$);

Output:
Define predictions through the final strong classifier result and test dataset ($H(a_i) = \mathrm{sign}(\sum_{t=1}^{T} a_t h_t(a_i))$).

Theoretically and empirically, this algorithm is a powerful ensemble learning algorithm [68]. First, this algorithm is trained through the discovered process model. Hence, the algorithm trains weak classifiers constantly. Weak classifiers focus on samples mislabeled in the previous steps. Therefore, process model execution is evaluated using the trained algorithm.

After controlling the behavior of executing processes, smart factory performance criteria are calculated based on KPI in the process instance controller function. Calculating these criteria is necessary to evaluate the performance of the optimization process.

3-3-1-2 Process instance controller function

This function inward control module can control the execution of process instances based on real-time data of execution through BPI techniques. BPI can measure the activities of a company and show the optimality and bottlenecks of processes [69]. Smart factory performance indicators are calculated based on runtime data. As shown in Figure 2, the process evaluator is responsible for evaluating performance indicators. These indicators will use in the optimization module.

There are different evaluation criteria to evaluate different sorts of processes. In this paper, the process evaluator evaluates the process instances based on Eqs. (1), (2), and (3) [70]. These criteria have been identified and used following the KPI standard.

- **Time:** Process duration utility was generated from the activity flow, as follows [70].

$$Time = \frac{1}{n}\sum_{i=1}^{n}\frac{\left|\sum_{j=1}^{m}R\left(a_{i,j}\right)-Exp_i\left(ps\right)\right|}{Exp_i\left(ps\right)} \tag{1}$$

where n is the number of end products in the process *ps*, Exp_i is the expected duration for the ith product in *ps*, and $R(a_{i,j})$ is the real duration of the jth activity of the ith product in *ps*.

- **Cost**: The total cost-utility of the process [70].

$$\mathrm{Cos}t = \frac{\left|\sum_{i=1}^{z}P\cos t\left(ps,t_i\right)-P\cos t_e\left(ps\right)\right|}{P\cos t_e\left(ps\right)} \tag{2}$$

where z is the process cycle is divided into z parts and $P\cos t(ps,t_i)$ is the running cost of process *ps* in time t_i.

- **Quality**: Better process quality is achieved when the cost efficiency of transforming the process cost into the value of the external customers is high [70].

$$Quality = w \times (1 - CRP_i) + (1 - w) \times (1 - PA_i) \tag{3}$$

where CPR_i is the cost efficiency of a certain process that satisfies the external customers' demands, PA_i denotes the cost efficiency of the process in supporting the primary activities to achieve their targets, and W is a lower weighting (w) when there is more concern about customer satisfaction.

3-3-2 Optimization Module

As mentioned earlier, the process should be optimized when there is a difference between the running process model and the model extracted from log files. In this paper, a BPI approach is used for process optimization to identify and evaluate the effectiveness of BPs. In this module, process-related rules are retrieved from the rules engine, and the appropriate values are placed in the process attributes. Regarding the multiplicity of BPs and the high volume of data resulting from implementing processes, determining the overall structure of BPI depends on the expected output of organizational managers. In this paper, analysis and optimization of production processes are considered. The parts used for optimizing the process model are shown in Figure 2.

1- **Change rule**: In this step, the defined rules of the process are retrieved from the Rules Engine module and are applied to the process model.
2- **Metric analysis**: The evaluation of smart factory performance indicators is recalculated after applying new rules to the process model.

Optimizing process models will improve KPI indicators and ultimately enhance the satisfaction of organizational managers.

4- Performance evaluation

The proposed architecture was evaluated by considering a medium-sized food production smart factory. The plant uses industry 4.0 technologies, including IoT and collaborative robots. Event processing and production management are performed in real-time. The factory structure is designed based on a 3-layer model. The bottom layer, called the physical layer, includes all the physical devices. The next layer, called the data layer, transfers data from machines to higher layers. Finally, the highest layer, i.e., the control layer, controls and optimizes the process.

For evaluation, we first simulated a virtual prototype of the smart factory using Digital Twin technology. Digital Twin is a virtual set of infrastructures and facilities that simulate a product in industry and factories. We used DC-E DigitalClone software to simulate the conditions of the smart factory.

4-1 Evaluation of the monitoring component

Control and optimization modules' inward monitoring component was evaluated separately in the following subsections.

4-1-1 evaluation of the control module

The performance of the control module was assessed by applying Python programming language and the ProM 6.10 tool. Several cases were tested through the latest ProM version. This version uses extended versions of plug-ins, which can provide better results than other versions. Also, Hadoop 3.3.0, and Spark 3.3.1 were used in these experiments. Experiments were performed on a computer with a triple-core processor, 12 GB of RAM, and 1 TB of hard disk space. The features of the log files are presented in Table 5.

4-1-1-1 Evaluation metrics

The process model discovery algorithm was evaluated using fitness functions based on Eqs. (4) to (8) [66]:

$$Completeness = \frac{\text{all parsed activites of casual matrix} - \text{penalty}}{\text{number of event log activites}} \tag{4}$$

$$Penalty = \frac{\text{all mising relations of casual matrix}}{\text{number of traces log} - \text{number of traces missing relation} + 1} \tag{5}$$

$$\mathrm{Pr}ecision = 1 - \max\{0, P_d, P_r\} \tag{6}$$

$$P_d = \frac{1}{\text{all enabled activites of the discovered model}} \tag{7}$$

$$P_r = \frac{1}{\text{all enabled activites of the real model}} \tag{8}$$

Also, the conformance-checking algorithm was evaluated through the following concepts and metrics with Eqs. (9) to (12) [7,66]:

- True Positive: process instances successfully detected as correct instances;
- False Negative: process instances predicted as incorrect instances but should be detected as correct instances;
- True Negative: process instances successfully detected as incorrect instances; and

- False Positive: process instances predicted as correct instances but should be detected as incorrect instances

$$\text{True positive rate} = \frac{\#\text{ true positive}}{\#\text{ true positive} + \#\text{ false negative}} \tag{9}$$

$$\text{False positive rate} = \frac{\#\text{ false positive}}{\#\text{ false positive} + \#\text{ true negative}} \tag{10}$$

$$\text{Accuracy} = \frac{\#\text{ true negative} + \#\text{true positive}}{\text{N}} \tag{11}$$

$$N = \#\text{ true negative} + \#\text{ false negative} + \#\text{ true positive} + \#\text{ false positive} \tag{12}$$

4-1-1-2 Evaluation results of the control module

To compare the performance of the NSGA-II algorithm with other algorithms, Tables 6, 7, and 8 show the list of parameters selected for them. Also, Table 9 shows the evaluation results of the process discover step based on the log files characteristics presented in Table 5.

In general, evaluating the metrics revealed that NSGA-II and Inductive Miner are more suitable mining algorithms. Inductive Miner provided a better completeness value than NSGA-II, but its precision is less than NSGA-II. As a result, new traces may be found in the discovered model that is not seen in the event log. Therefore, if the discovered process is not accurate enough, the evaluation of deviations in it will not be accurate. Hence, a more accurate algorithm is more important in our work. Also, as shown in Table 9, when a trade-off between completeness and precision is required, and the overall quality of the discovered model is important, the NSGA-II algorithm is a more reliable option. On the other hand, a conventional process discovery algorithm produces a single process model at a time, but NSGA-II can simultaneously produce a set of models by constructing a Pareto front process model. In this way, the user can choose a process model with a trade-off between quality dimensions based on his/her preferences. Therefore, based on a general evaluation, it can be concluded that the NSGA-II provides better performance in discovering process models.

Table 10 shows the evaluation results of the process conformance check step based on the log files characteristics presented in Table 5.

Table 10 presents the robustness and optimal performance of the AdaBoost algorithm to conform process models based on different logs. Obviously, the high accuracy of this algorithm is the main reason for obtaining optimal results.

4-1-2 evaluation of the optimization module

As mentioned in Section 3, optimization occurs when there is a difference between the executable model and the model extracted from the log files. Process instance optimization is performed at the time of process execution. Therefore, the optimization module must be evaluated in real conditions.

4-1-2-1 Evaluation metrics

In a smart factory, each process follows some rules, including threshold values based on standard KPI. These thresholds are about the features of the process, such as cost, time, and quality. In this study, some rules are considered about the production processes as follows:

- **Rule 1)** The maximum total execution time of a production process is 200 ms. If the total execution time of the executable model is more than 200 ms, set a lower execution priority for the process.
- **Rule 2)** The minimum total execution time of a production process is 100 ms. If the total execution time of the executable model is less than 100 ms (i.e., the process failure), put the process in the process queue for the next execution.

4-1-2-2 Evaluation results of the optimization module

After running the proposed BPMS in the simulation environment, event logs are collected in log files. If the executable model is different from the model extracted from log files, the related rule must be fetched from the rules engine, and appropriate values must be applied to the process. Table 11 presents the result of running this procedure in the virtual smart factory.

Table 11 compares all process models with models extracted from log files based on the total execution time criterion. If the total execution time of the executable model is different from the model extracted from the log file, the rules related to the total execution time are called and applied to the process to obtain the desired result at the execution time of the processes.

5- Discussion

With the advent of smart factories and the need to use the latest technologies, such as the IoT, big data, and cloud computing, BPM has become increasingly essential. However, one of the major challenges is that existing BPMSs is not optimized enough to be used in today's industries with high data volumes, high dynamicity, and high scalability. In this regard, classic BPMSs have a limited ability to execute BPs in a dynamic environment such as smart factories. Therefore, this paper presents a new architecture of a BPMS for smart factories. In the proposed architecture, the integration of Apache Big Data Analysis tools and process mining techniques has been used for real-time processing. In this way, big data processing is done with considerable speed, and log files are provided to other modules more efficiently.

Since the elements used in BPMN 2.0 are not sufficient for modeling dynamic IoT-based processes, this paper developed a new extension of the BPMN 2.0 language with accepted new elements of reference [64] in the modeling component. The new extension makes it possible to design dynamic business models more accurately and in accordance with the dynamic nature of smart factories.

As mentioned in the article, the conditions of smart factories should be monitored according to their changing status. These continuous changes are not sufficiently considered in most existing BPMSs. Therefore, the proposed architecture includes a monitoring component. This component comprises a control module to discover process models and their conformity with models extracted from BP analysis. The NSGA-II algorithm is used to discover the process models, and the AdaBoost algorithm is used to check the conformity of the process models. The monitoring component also includes an optimization module to improve the BP model based on the BPI approach and KPIs. Overall, the ability of the optimization module to change the BP rules is another advantage of the proposed architecture over other BPMSs.

In the control module, the NSGA-II algorithm is evaluated based on completeness and precision criteria. Regarding its significant advantages, such as non-penalty constraint handling, it outperforms other algorithms presented in Table 9. The AdaBoost algorithm was evaluated by examining the false positive rate, false negative rate, and accuracy criteria. Based on the results obtained in Table 10, this algorithm has high accuracy and a low error rate. The remarkable point in this regard is that the AdaBoost algorithm has adapted process models more accurately and quickly after using big data solutions.

The process models are improved in the optimization module using the BPI technique. As mentioned, to increase flexibility, a BPMS must be able to change the BP rules. This requirement was met in the optimization module. The evaluation results of the module in Table 11 indicate the dynamics of the proposed system in different conditions.

Based on the suggested solutions, the proposed architecture is suitable for implementing in a smart factory with IoT and big data technologies. In fact, the proposed architecture led to further aligning the business with IT. In addition, it will be possible for managers to achieve competitive conditions in the labor market, respond quickly to customers, and response to a wide range of changes. However, some constraints in this paper have not been considered. Given the uncertain nature of business conditions, considering the uncertainty of BPs is a challenging issue that has not been addressed in this study. In addition, due to the variable conditions of the smart factory, the management of the rules stored in the rules engine is essential for managers and supervisors. Therefore, changing the threshold values in the rules requires a method not mentioned in this study.

6- Conclusions and Future works

This paper presented an extended BPMS architecture prototype to monitor a smart factory's processes. As illustrated earlier, process characteristics in this type of factory are based on new IT technologies (e.g., big data, IoT, and cloud computing) and in a smart status. Also, due to the dynamic nature of the processes in smart factories, different systems and tools (e.g., BPMS) should be applied to control processes in dynamic conditions. In this regard, classic BPMSs have a limited ability to execute BPs in a dynamic environment such as smart factories. Therefore, based on the presented research questions, to answer the first research question (Rq1), this study developed a new architecture of a BPMS by combining existing architectures with functions provided by modern techniques such as process mining and big data analytics for smart factories. Also, the second research question (Rq2) has been answered in the way that in the proposed architecture, the integration of Apache Big Data Analysis tools and process mining techniques has been used for processes monitoring. This paper mainly attempted to propose a robust BPMS architecture related to the smart factory features. This prototype expands the current BPMS architectures by:

- IoT-based dynamic processes modeler according to the environmental conditions;
- A defined data analysis system for gathering event logs from big data;
- Monitoring component that can control the behavior of processes using control and optimization modules inside powerful algorithms; and
- The improvement in control of dynamic processes due to using process mining techniques.

For future works, we envisaged several appropriate research directions as follows:

- Employing a new component in the proposed architecture to investigate security issues and prevent possible attacks in cyber-physical environments;
- Defining interoperability attribute inward proposed architecture to reuse the components of this architecture in other architectures;
- Presenting the new proposed architecture through a user-friendly graphical user interface with usability for non-technical end users and appropriate software development standards.

References

[1] Rossit, D. and Tohmé, F. "Scheduling research contributions to Smart manufacturing", *Manufacturing Letters*, **15**, pp. 111-114 (2018).

[2] Weske, M. "Business Process Management-Concepts, Languages, Architectures", 2th Edn., Springer, Berlin, Heidelberg (2012).

[3] van der Aalst, W. M. P., La Rosa, M., and Santoro, F. M. "Business process management-Don't forget to improve the process!", *Business & Information Systems Engineering*, **58**, pp.1-6 (2016).

[4] Dumas, M., La Rosa, M., Mendling, J., et al. "Fundamentals of business process management", 1th Edn., Springer Berlin, Heidelberg (2013).

[5] Chinosi, M. and Trombetta, A. "BPMN: An introduction to the standard", *Computer Standards & Interfaces*, **34**(1), pp. 124-134 (2012).

[6] Delgado, A., Calegari, D., and Arrigoni, A. "Towards a generic BPMS user portal definition for the execution of business processes", *Electronic Notes in Theoretical Computer Science*, **329**, pp. 39-59 (2016).

[7] van der Aalst, W. "Process Mining: Data Science in Action", 2th Edn., Springer Berlin Heidelberg (2016).

[8] Badakhshan, P., Wurm, B., Grisold, T., et al. "Creating business value with process mining", *The Journal of Strategic Information Systems*, **31**(4), pp. 101745 (2022).

[9] Halaška, M. and Šperka, R. "Process mining–the enhancement of elements industry 4.0", In *2018 4th International Conference on Computer and Information Sciences (ICCOINS)*, Kuala Lumpur, Malaysia, pp. 1-6 (2018).

[10] Lee, J., Ardakani, H. D., Yang, S., et al. "Industrial big data analytics and cyber-physical systems for future maintenance & service innovation", *Procedia cirp*, **38**, pp. 3-7 (2015).

[11] Perera, C., Liu, C. H., Jayawardena, S., et al. "A survey on internet of things from industrial market perspective", *IEEE Access*, **2**, pp.1660-1679 (2014).

[12] Xu, X. "From cloud computing to cloud manufacturing", *Robotics and computer-integrated manufacturing*, **28**(1), pp. 75-86 (2012).

[13] Monostori, L., Kádár, B., Bauernhansl, T., et al. "Cyber-physical systems in manufacturing", *Cirp Annals*, **65**(2), pp. 621-641(2016).

[14] Vasilecas, O., Kalibatiene, D., and Lavbič, D. "Rule-and context-based dynamic business process modelling and simulation", *Journal of Systems and Software*, **122**, pp. 1-15 (2016).

[15] Cimini, C., Pinto, R., and Cavalieri, S. "The business transformation towards smart manufacturing: a literature overview about reference models and research agenda", *IFAC-PapersOnLine*, **50**(1), pp. 14952-14957 (2017).

[16] Chapela-Campa, D., Mucientes, M., and Lama, M. "Mining frequent patterns in process models", *Information Sciences*, **472**, pp. 235-257 (2019).

[17] Alizadeh, S. and Norani, A. "ICMA: A new efficient algorithm for process model discovery", *Applied Intelligence*, **48**, pp. 4497-4514 (2018).

[18] van der Aalst, W., Weijters, T., and Maruster, L. "Workflow mining: Discovering process models from event logs", *IEEE transactions on knowledge and data engineering*, **16**(9), pp. 1128-1142 (2004).

[19] De Medeiros, A. K. A., Van Dongen B. F., Van der Aalst W.M., et al. "Process mining: extending the alpha-algorithm to mine short loops", Technische Universiteit Eindhoven, Eindhoven (2004).

[20] Weijters, A. J. M. M. and Van der Aalst, W. M. "Rediscovering workflow models from event-based data using little thumb", *Integrated Computer Aided Engineering*, **10**(2), pp. 151-162 (2003).

[21] de Medeiros, A. K. A., Weijters, A. J., and van der Aalst, W. M. "Genetic process mining: an experimental evaluation", *Data Mining and Knowledge Discovery*, **14**, pp. 245-304 (2007).

[22] Chifu, V. R., Pop, C. B., Salomie, I., et al. "Hybrid particle swarm optimization method for process mining", In *2012 IEEE 8th International Conference on Intelligent Computer Communication and Processing*, Cluj-Napoca, Romania, pp. 273-279 (2012).

[23] Xu, F., Yang, J., Liu, L., et al. "A Method of Process Mining based on Genetic Algorithm", In *2021 3rd International Academic Exchange Conference on Science and Technology Innovation (IAECST),* Guangzhou, China, pp. 698-703, (2021).

[24] Berti, A. and van der Aalst, W. M. "A novel token-based replay technique to speed up conformance checking and process enhancement", *In Transactions on Petri Nets and Other Models of Concurrency XV*, pp. 1-26 (2021).

[25] Burattin, A., Zelst, S. J. V., Armas-Cervantes, A., et al."Online conformance checking using behavioural patterns", In *International Conference on Business Process Management*,**16**, Sydney, NSW, Australia, pp. 250-267 (2018).

[26] Leoni, M. D., Van Der Aalst, W. M., and Dongen, B. F. V. "Data-and resource-aware conformance checking of business processes", In *International Conference on Business Information Systems*, **15**, Vilnius, Lithuania, pp. 48-59 (2012).

[27] Tiwari, A., Vergidis, K., and Majeed, B. “Evolutionary multi-objective optimization of business processes”, In *2006 IEEE International Conference on Evolutionary Computation*, Vancouver, BC, Canada, pp. 3091-3097 (2006).

[28] Saraeian, S., Shirazi, B., and Motameni, H.”Adaptive control of criticality infrastructure in automatic closed-loop supply chain considering uncertainty”, *International Journal of Critical Infrastructure Protection*, **25**, pp. 102-124 (2019).

[29] Vera-Baquero, A., Colomo-Palacios, R., and Molloy, O. “Business process analytics using a big data approach”, *It Professional*, **15**(6), pp. 29-35 (2013).

[30] Pourmirza, S., Peters, S., Dijkman, R., et al. “BPMS-RA: a novel reference architecture for business process management systems”, *ACM Transactions on Internet Technology (TOIT)*, **19**(1), pp. 1-23 (2019).

[31] Krumeich, J., Werth, D., and Loos, P. “Prescriptive control of business processes”, *Business & Information Systems Engineering*, **58**, pp. 261-280 (2016).

[32] Khan, R., Azam, F., Maqbool, B., et al.”A Framework for automated reengineering of BPMN models by excluding inefficient activities”, In *Proceedings of the 2020 9th International Conference on Software and Computer Applications*, Langkawi, Malaysia, pp. 147-151 (2020).

[33] Durán, F., Martínez, D., and Rocha, C.” Business Processes Analysis with Resource-Aware Machine Learning Scheduling in Rewriting Logic”, In *International Workshop on Rewriting Logic and its Applications*, Munich, Germany, pp. 113-129 (2022).

[34] Vergidis, K. and Tiwari, A. “Business process design and attribute optimization within an evolutionary framework”, In *2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)*, Hong Kong, China, pp. 668-675 (2008).

[35] Zhao, W., Zeng, Q., Zheng, G., et al. “The resource allocation model for multi-process instances based on particle swarm optimization”, *Information Systems Frontiers*, **19**, pp. 1057-1066 (2017).

[36] Mahammed, N., Fahsi, M., and Bennabi, S.” Genetic Algorithm Based on Multiple Population in a Business Process Optimization Issue”, In *2020 International Conference on Advanced Aspects of Software Engineering (ICAASE)*, Constantine, Algeria, pp. 1-6 (2020).

[37] Jiang, L., Wang, J., Shah, N., et al. “A process-mining-based scenarios generation method for SOA application development”, *Service Oriented Computing and Applications*, **10**, pp. 303-315 (2016).

[38] Yang, H., Park, M., Cho, M., et al. “A system architecture for manufacturing process analysis based on big data and process mining techniques”, In *2014 IEEE international conference on big data (big data)*, Washington, DC, USA, pp. 1024-1029 (2014).

[39] Hongtao, T., Yong, C., and Jiansa, L. U. “Architecture of process mining based business process optimization”, *International Technology and Innovation Conference 2006 (ITIC 2006)*, Hangzhou, China, p.1066-1069 (2006).

[40] Hollingsworth, D. "Workflow management coalition: The workflow reference model", Document Number TC00-1003 (1995).

[41] Grefen, P. and de Vries, R. R. "A reference architecture for workflow management systems", *Data & Knowledge Engineering*, **27**(1), pp. 31-57 (1998).

[42] Arsanjani, A., Zhang, L. J., Ellis, M., et al. "S3: A service-oriented reference architecture", *IT professional*, **9**(3), pp. 10-17 (2007).

[43] Lin, C., Lu, S., Fei, X., et al. "A reference architecture for scientific workflow management systems and the VIEW SOA solution", *IEEE Transactions on Services Computing*, **2**(1), pp. 79-92 (2009).

[44] Ladleif, J., Friedow, C., and Weske, M. "An architecture for multi-chain business process choreographies", In *Business Information Systems: 23rd International Conference, BIS 2020,* **23**, Colorado, USA, pp. 184-196 (2020).

[45] Junying, X., Yan, L., and Ziyu, L. "Research and implementation on communication organization workflow management system based on service-oriented architecture", In *2020 IEEE International Conference on Artificial Intelligence and Information Systems (ICAIIS),* Dalian, China, pp. 548-551 (2020).

[46] Ahmad, Z., Nazir, B., and Umer, A. "A fault- tolerant workflow management system with Quality- of- Service- aware scheduling for scientific workflows in cloud computing", *International Journal of Communication Systems*, **34**(1), pp. e4649 (2021).

[47] Ouyang, C., Adams, M., Hofstede, A. H. T., et al. "Design and Realisation of Scalable Business Process Management Systems for Deployment in the Cloud", *ACM Transactions on Management Information Systems (TMIS)*, **12**(4), pp. 1-26 (2021).

[48] Chaudhry, N. R., Anagnostou, A., and Taylor, S. J." A workflow architecture for cloud-based distributed simulation", *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, **32**(2), pp. 1-26 (2022).

[49] Flechsig, C., Völker, M., Egger, C., et al. "Towards an integrated platform for business process management systems and robotic process automation", In *International conference on business process management,* Munster, Germany, pp. 138-153 (2022).

[50] Meidan, A., García-García, J. A., Escalona, et al. "A survey on business processes management suites", *Computer Standards & Interfaces*, **51**, pp. 71-86 (2017).

[51] Wörzberger, R., Ehses, N., and Heer, T. "Adding support for dynamics patterns to static business process management systems", In *International Conference on Software Composition,* **7**, Budapest, Hungary, pp. 84-91 (2008).

[52] Saraeian, S., Shirazi, B., and Motameni, H. "Towards an extended BPMS prototype: Open challenges of BPM to flexible and robust orchestrate of uncertain processes", *Computer Standards & Interfaces*, **57**, pp. 1-19 (2018).

[53] Schulte, S., Janiesch, C., Venugopal, S., et al. "Elastic Business Process Management: State of the art and open challenges for BPM in the cloud", *Future Generation Computer Systems*, **46**, p. 36-50 (2015).

[54] Alexopoulou, N., Nikolaidou, M., Anagnostopoulos, D., et al. "An event-driven modeling approach for dynamic human-intensive business processes", *In International Conference on Business Process Management,* Ulm, Germany, pp. 393-404 (2009).

[55] Kozma, D., Varga, P., and Larrinaga, F. "Dynamic multilevel workflow management concept for industrial IoT systems", *IEEE Transactions on Automation Science and Engineering*, **18**(3), pp. 1354-1366 (2020).

[56] Seiger, R., Malburg, L., Weber, B., et al. "Integrating process management and event processing in smart factories: A systems architecture and use cases", *Journal of Manufacturing Systems*, **63**, pp. 575-592 (2022).

[57] Gorski, E. G., Loures, E. D. F. R., Santos, E. A. P., et al. "Towards a smart workflow in CMMS/EAM systems: An approach based on ML and MCDM", *Journal of Industrial Information Integration*, **26**, pp.100278 (2022).

[58] Garcia Represa, J. and Delsing, J. "Manufacturing workflows in microservice architectures supporting digital transactions for business process automation", In *Intelligent and Transformative Production in Pandemic Times: Proceedings of the 26th International Conference on Production Research,* pp. 113-126 (2023).

[59] Li, Y., Goga, K., Tadei, R., et al. "Production scheduling in Industry 4.0", In *Complex, Intelligent and Software Intensive Systems: Proceedings of the 14th International Conference on Complex, Intelligent and Software Intensive Systems (CISIS-2020),* lodz, poland, pp. 355-364 (2021)

[60] Schönig, S., Ackermann, L., Jablonski, S., et al. "IoT meets BPM: a bidirectional communication architecture for IoT-aware process execution", *Software and systems modeling*, **19**, pp. 1443-1459 (2020).

[61] Malburg, L., Hoffmann, M., and Bergmann, R. "Applying MAPE-K control loops for adaptive workflow management in smart factories", *Journal of Intelligent Information Systems*, **61**, pp. 1-29 (2023).

[62] Ochoa, W., Larrinaga, F., and Pérez, A. "Architecture for managing AAS-based business processes", *Procedia Computer Science,* **217**, pp. 217-226 (2023).

[63] Saraeian, S. and Shirazi, B. "The role of bi-level uncertain architecture inward smart manufacturing: Process orchestration", *Scientia Iranica,* **30**(6), pp. 2098-2115 (2023).

[64] Meyer, S., Sperner, K., Magerkurth, C., et al. “Internet of Things Architecture-IoT-A-Project Deliverable D2. 2–Concepts for Modelling IoT-Aware Processes”, Doc. Ref.: D2. 2 (2012).

[65] Agarwal, M., Gupta, S., and Kumar, N. “Discovering pareto-optimal process models: a comparison of MOEA techniques”, *In Proceedings of the Genetic and Evolutionary Computation Conference Companion,* Kyoto, Japan, pp. 286-287 (2018).

[66] Saraeian, S. and Shirazi, B. “Process mining-based anomaly detection of additive manufacturing process activities using a game theory modeling approach”, *Computers & Industrial Engineering*, **146**, pp. 106584 (2020).

[67] Tsai, C. Y., Jen, H., and Chen, Y. C. “Time-interval process model discovery and validation—a genetic process mining approach”, *Applied Intelligence*, **33**, pp. 54-66 (2010).

[68] Sun, B., Chen, S., Wang, J., et al. “A robust multi-class AdaBoost algorithm for mislabeled noisy data”, *Knowledge-Based Systems*, **102**, pp. 87-102 (2016).

[69] Linden, M., Felden, C., and Chamoni, P. “Dimensions of business process intelligence”, *In International Conference on Business Process Management*, Hoboken, NJ, USA, pp. 208-213 (2010).

[70] Tan, W., Shen, W., Xu, L., et al. “A Business Process Intelligence System for Enterprise Process Performance Management”, *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, **38**(6), pp. 745-756 (2008).

Figures captions:

Figure 1- Workflow management system

Figure 2- The proposed BPMS architecture

Figure 3- Metamodel of the integration of BPMN 2.0 version and extended version

Figure 4- Data analysis method

Figure 5- Description of process model controller

Figure 6- The flowchart of the NSGA-II for process model discovery

Tables captions:

Table 1- A review of BPMS architectures characteristics

Table 2- A review of Commercial BPMSs

Table 3- An overview of some of BPMSs in smart factories

Table 4- New extension elements of the BPMN language

Table 5- Features of the log files

Table 6- Selected parameters for the genetic algorithm

Table 7- Selected parameters for the heuristic algorithm

Table 8- Selected parameters for the NSGA-II algorithm

Table 9- Comparison of the NSGA-II algorithm in the process discovery step with other algorithms

Table 10- Validation results of the AdaBoost algorithm in process conformance step

Table 11- Results of implementing the optimization module

Table 1- A review of BPMS architectures characteristics

References	**Architecture Characteristics**
[43]	The first reference architecture for scientific WFMS is presented using SOA features.
[30]	A defined reference architecture for the BPMS entitles BPMS-RA intending to integrate real-time analysis of BP inward the proposed architecture
[44]	A developed BPMS architecture based on blockchain technology for use in multi-chain environments.
[45]	A workflow that is defined by combining workflow and web service technologies. This workflow is designed based on SOA.
[46]	The architecture of the QoS-aware fault-tolerant workflow-based system is employed in the cloud computing environments.
[47]	A scalable BPMS architecture for deployment in the cloud computing environment
[48]	A new workflow architecture for distributed simulation on cloud.
[49]	A presented architecture called BPMS-RPA based on the integration of robotic process automation technology in a BPMS

BPMS name	Modeling language	Design (supported programming language)	Deployment metrics	Monitoring & Control	Analysis metrics
Bonita	BPMN2.0	Java	Ability to integrate into	Technical monitoring and control, changing	Process verification and simulation

			other services and systems	role, resource, and workload balance	
Activiti	BPMN2.0	Java and JavaScript	Ability to integrate with other services and systems	Changing role, resource, and workload balance	Process verification and simulation
jBPM	BPMN2.0	Java	Distributed execution and ability to integrate with other services and systems	Changing role or resource	Process verification and simulation; Using historical data for analysis
Process Maker	BPMN2.0	JavaScript and PHP	Ability to integrate with other services and systems	Changing role, resource, and workload balance	Using historical data for analysis through events log
uEngine BPM	XPDL	Java	Ability to integrate with other services and systems	Business monitoring and control	Using historical data for analysis
YAWL	YAWL	Java	Ability to integrate with other services and systems	Changing role, resource, and workload balance	Process verification and simulation; Using historical data for analysis
Camunda	BPMN2.0	Java and JavaScript	Ability to integrate with other services and systems	Changing role, resource, and workload balance	Process verification and Using historical data for analysis

Table 2- A review of Commercial BPMSs [50]

Table 3- An overview of some of BPMSs in smart factories

References	Issue/challenges	Proposed solution	Results

[60]	Delay in reaction time in terms of interaction between users and machinery cause low-performance products	Presenting an architecture to collect data of sensors and apply them in BPMS using IoT	Production optimization, reducing reaction time to tasks execution, and increase tasks execution quality
[61]	Monitoring the status of running processes and reacting to conditions that occur during runtime	Proposing an architecture by employing MAPE-K (Monitor, Analyze, Plan, Execute, and Knowledge) control loops for adaptive WFMS in smart factories	Identify failures and resolve them autonomously
[62]	The need to orchestrate devices' services to manage unpredictable conditions in the manufacturing system	Providing an architecture to manage business processes based on Asset Administration Shell (AAS)	Orchestration of device services by business processes and AAS and as a result more interoperability between manufacturing systems
[63]	Using uncertain processes to carry out smart productions in smart factories	Presenting a developed uncertain BPMS architecture to manage uncertain processes in the smart factory	Orchestration of services provided by uncertain business processes in the field of smart productions

Table 4- New extension elements of the BPMN language
(adapted from [64])

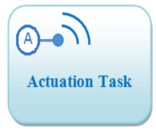
Element	Symbol	Description
Actuation Task		Data or commands that are sent to an actuation device
Sensing Task		Data that is read or received by a sensor device
Mobility		A process, action, or activity performed by a mobile device
Real World Data Object		A data object used by an IoT device
Real World Data Store		Storage of data, such as a repository to collect sensor data
IoT Device		An IoT device includes its components, e.g., sensor and actuator.

Table 5- Features of the log files

Event Log Name	Description	Size on Disk	Number of Cases	Number of Events
Log1 of the Food Production	Data about the Food Production System	72.8 MB	20135	309036
Log2 of the ERP	Data of an ERP System	4.84 MB	34723	103469
Log3 of the SCM	Data of Supply Chains containing structured data	91.1 MB	20652	180519
Log4 of the SCM	Data of Supply Chains containing unstructured data	91 MB	3340	469977

Table 6- Selected parameters for the genetic algorithm

Population size	Generation	Extra behavior punishment	Mutation Probability	Crossover Probability
10	100	0.025	0.2	0.8

Table 7- Selected parameters for the heuristic algorithm

Long distance	Length two loops	Loops length one	Dependency
0.9	0.9	0.9	0.9

Table 8- Selected parameters for the NSGA-II algorithm

Population size	Generation	Mutation Probability	Crossover Probability
10	100	0.2	0.8

Table 9- Comparison of the NSGA-II algorithm in the process discovery step with other algorithms (log1, log2, log3, and log4 are the algorithm inputs)

Algorithm Name	Completeness (%)	Precision (%)
Alpha	61.4	79.34
	overall: 70.37	
Genetic	0.78	0.48
	overall: 0.63	
Heuristic	25.42	52.99
	overall: 39.20	
Inductive Miner	85.64	75.51
	overall: 80.57	
NSGA-II (This Paper)	80.29	100
	overall: 90.14	

Table 10- Validation results of the AdaBoost algorithm in process conformance step
(log1, log2, log3, and log4 are as the algorithm inputs)

Event Log Name	True Positive Rate (%)	False Positive Rate (%)	Accuracy (%)	Iteration
Log1	100	10.94	98.26	64
Log2	97.6	0	97.2	61
Log3	99.99	20.8	0.99	65
Log4	100	20.92	99.08	32

Table 11- Results of implementing the optimization module

Process ID	Total Execution Time (Executed Model)	Total Execution Time (Log File Extracted Model)	Need Optimization?	Optimization Result (Executed Model)
P#34	215	198	✓	198
P#42	155	155	-	-
P#59	94	115	✓	115

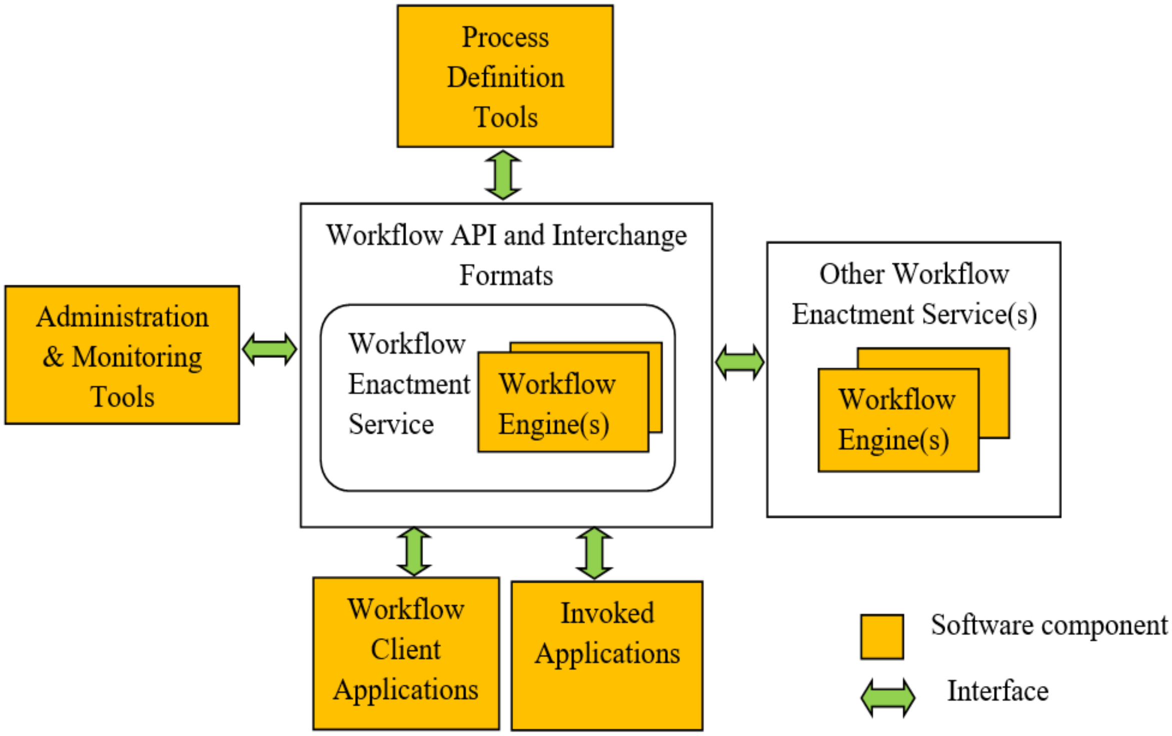


Figure 1- Workflow management system [40]

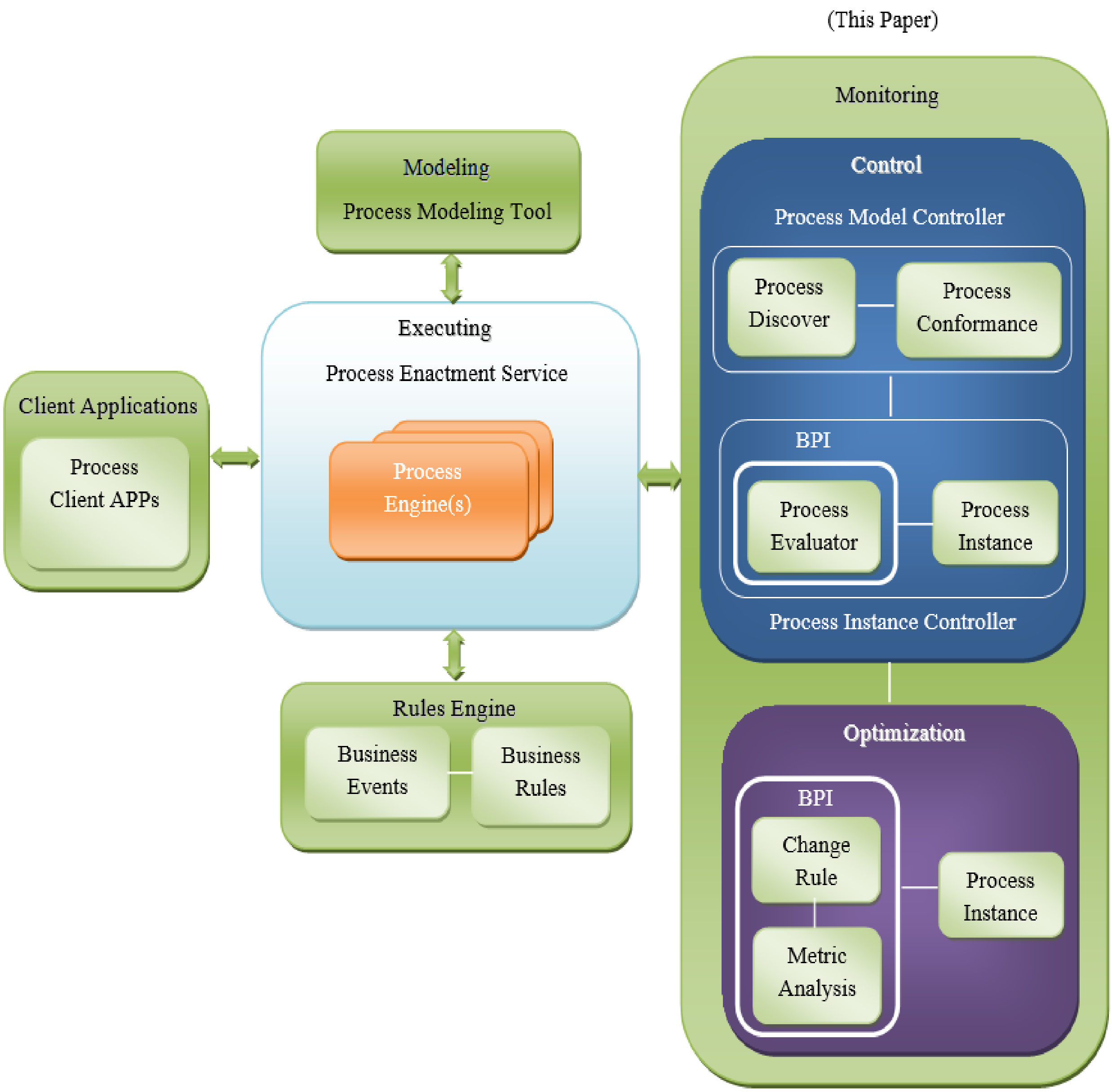


Figure 2- The proposed BPMS architecture

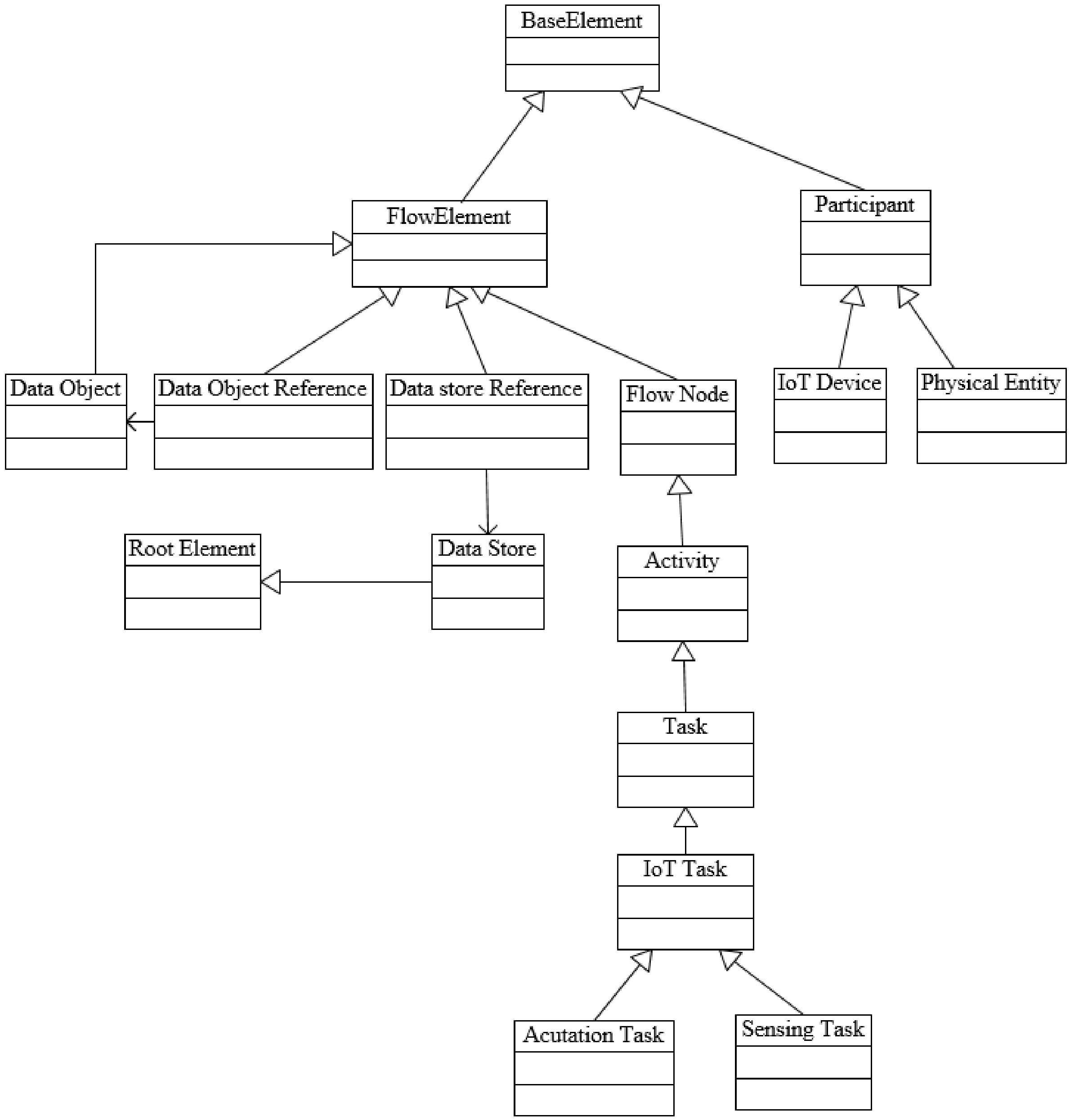


Figure 3- Metamodel of the integration of BPMN 2.0 version and extended version

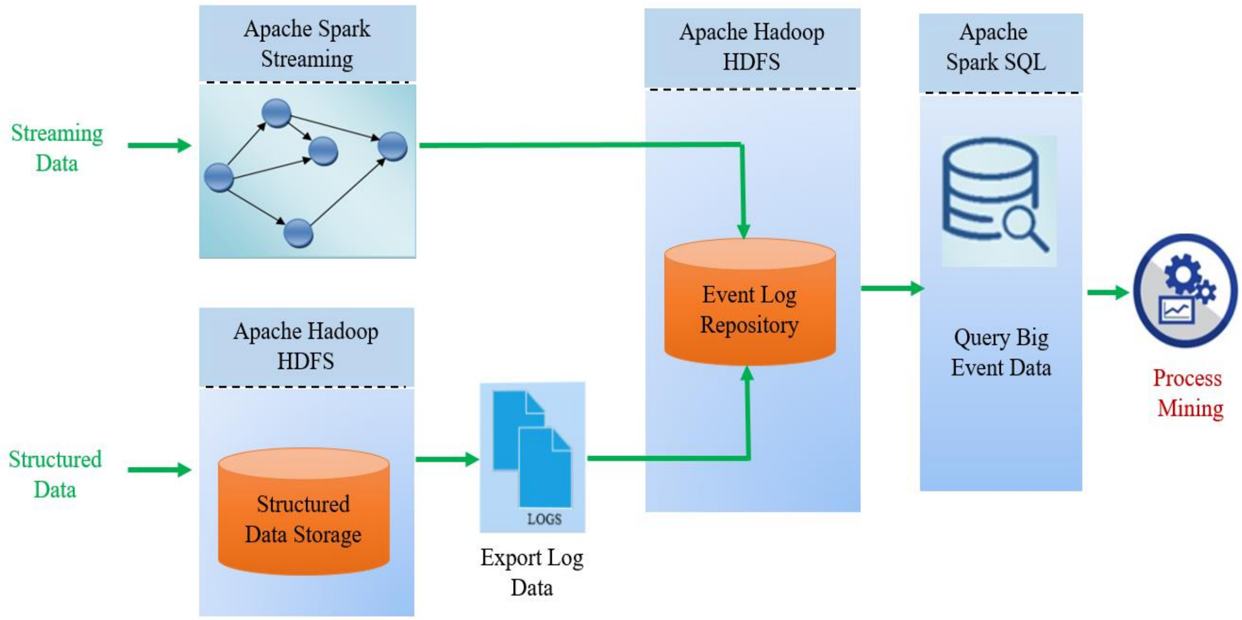


Figure 4- Data analysis method

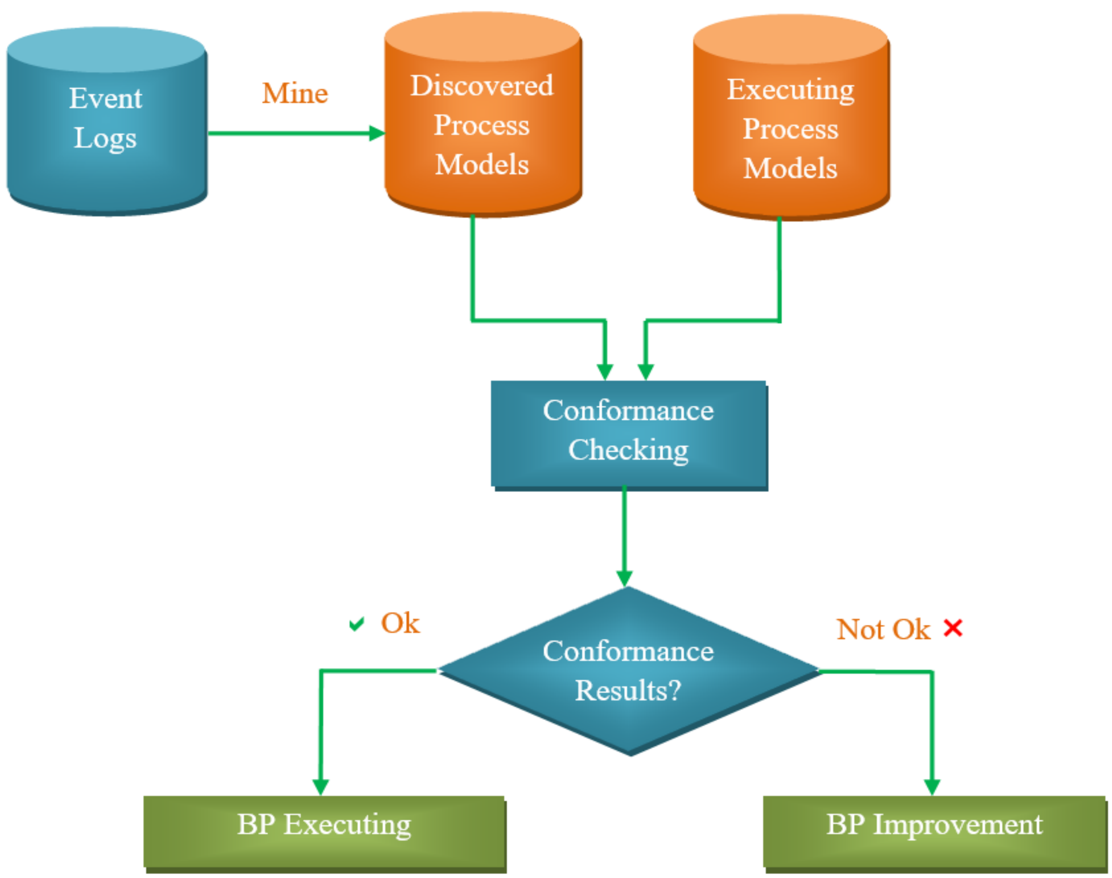


Figure 5- Description of process model controller

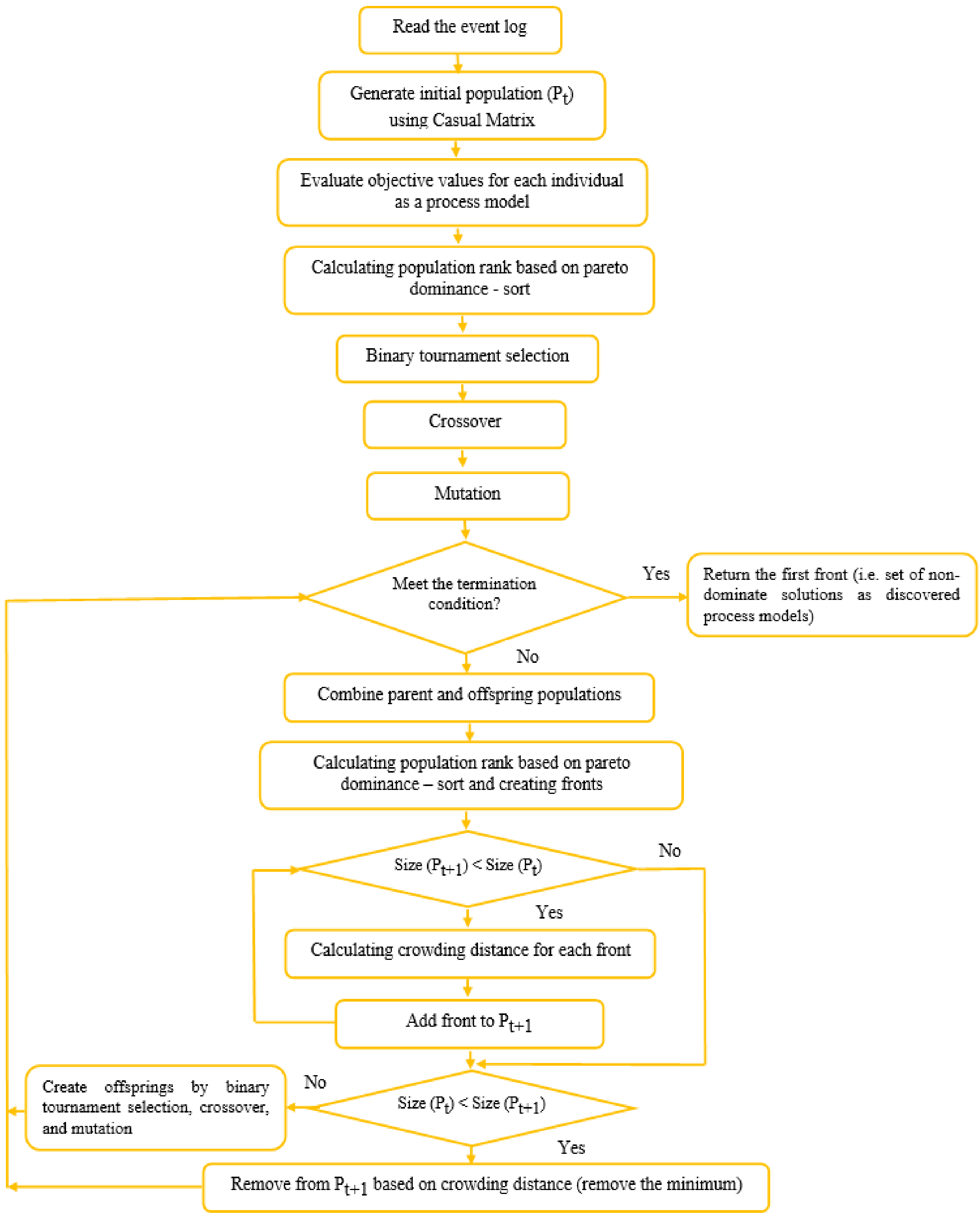


Figure 6- The flowchart of the NSGA-II for process model discovery

Biography

Alireza Olyai is PhD candidate of Computer Engineering at Gorgan Branch, Islamic Azad University, Gorgan, Iran. His research interests include software engineering, enterprise architecture, business process managent, process mininig, and big data analytics.

Shideh Saraeian is the assistant professor of Computer Engineering at Gorgan Branch, Islamic Azad University, Gorgan, Iran. She studied Computer Engineering in Najafabad Branch, Islamic Azad University, Esfahan, Iran from 2001 to 2005. She has graduated in MSc of Information Technology engineering from Yazd University, Yazd, Iran from 2006 to 2008. She has graduated in Ph.D of Computer Engineering in Sari Branch, Islamic Azad University, Sari, Iran from 2014 to 2017. Her research interests include uncertain Information Systems management, data/process miinig, machine learning/intelligent algorithms, and network security/anamoly detedction. She has published several research papers. She published several research papers. Also, she authored two books in Persain.

Ali Nodehi is an assistant professor in the Department of Computer Engineering at Gorgan Branch, Islamic Azad University, Gorgan, Iran. His research interests include soft computing, evolutionary algorithms and image processing.