Enhancing Business Process Performance Analysis through Coverage-based Monitoring

Antonello Calabrò, Francesca Lonetti, Eda Marchetti and Giorgio Oronzo Spagnolo
Istituto di Scienza e Tecnologie dell'Informazione "A. Faedo", CNR
Pisa, Italy
{firstname.lastname}@isti.cnr.it

***Abstract*—Currently, Business Process Model and Notation (BPMN) has becoming one of the main languages for creating a description of processes and developing executable frameworks for the management of the process itself. In this context, one important aspect of process management is the possibility of process analysis and optimization, which many times relies on real data collected during the process execution itself. This paper goes in this direction and presents a framework for performance analysis and optimization of the business process expressed in BPMN. It focuses on the collection and evaluation of time based and cost based parameters ad relies on an adaptable and flexible event based monitoring infrastructure. In particular, it implements some business process path coverage metrics useful for evidencing business process criticalities and possible improvements of the business process execution. A first validation on a case study is also presented.**

I. Introduction

Nowadays, more and more industrial organizations are using Business Process Model and Notation (BPMN) for process modeling. The main benefits of BPMN commonly rely on the possibility of having a simple and standard notation for creating a description of processes (in terms of participants and activities) and develop executable frameworks for the overall management of the process itself. Business Process (BP) analysis aims at clarifying the characteristics of a BP model, identifying its possible bottlenecks, and comparing any potential process alternatives.

Monitoring the BP execution represents a key aspect both for business process management, analysis and validation as evidenced by some of the recent works [1], [2], [3]. In particular, specific key performance indicators (KPIs), including time based and cost based parameters, are commonly adopted for controlling the BP and its specific goals. In literature there are several KPIs already defined for different purposes of business process analysis, such for instance those introduced in [4], which focus on the adequacy concepts. In this case the main goal is to define the entities (activities, connection objects, swimlanes, etc.) that we would expect to observe during BPMN execution and set adequacy criteria on them.

Data useful for KPIs measurement can be collected in different way [5], even if commonly monitoring facilities are used for the purpose. In this paper, we leverage on business process execution adequacy criteria defined in [4] so that they can be used to perform business process performance analysis and optimization. In this last case through a monitoring activity, the tasks or the set of tasks, that may cause bottleneck, delay or any other kind of performance violation can be identify and the BPMN model properly optimized.

From an implementation point of view, specific performance analysis metrics are designed for BP assessment including waiting time, synchronization time, path execution time. These metrics are assessed by using a flexible and adaptable monitoring framework based on *path extraction*. Data collected by this monitoring framework during the BP execution are analyzed and elaborated in order to calculate specific values for the defined performance metrics. The obtained values allow continuously tracking of the process behaviour as well as overall performance improving and optimization of the business process itself. The proposed approach has been applied to a case study, then performance analysis results of a business process and its optimization are presented in this paper.

The contribution of this paper can be summarized into: i) the definition of BP execution performance analysis metrics; ii) a proof-of-concept framework able to measure these performance analysis metrics; iii) a preliminary assessment of the proposed proof-of-concept framework on a case study showing optimization and evaluation of a business process.

The remainder of this paper is structured as follows: Section II addresses related works on performance analysis and monitoring of business process whereas Section III presents the proposed business process performance metrics and the monitoring framework used to assess them. Section IV presents a case study of the proposed approach. Finally, Section V concludes the paper also hinting at future work.

II. Related Work

Usually Business Process performance analysis focuses on the evaluation of the ability of the workflow to meet requirements with respect to some performance indicators such as throughput times, service levels, resource utilization or other quantitative factors. As highlighted in [6] Key Performance Indicators (KPI), Balanced Scorecard (BSC), and Business Excellence Model (BEM) are the more widely accepted and effective approaches for BP performance modeling and measurement. Over the last years conceptual frameworks have been proposed both for implementing such performance analysis approaches, such as [7], [8], [9], or for classifying the main techniques for BP modeling with regard to their analysis and optimization capabilities [10]. Results of such studies together with the analysis provided by [10] evidenced that Business Process Modeling Language (BPML) and Petri-nets are the most powerful modeling techniques supporting performance analysis.

Data collection for BP analysis is usually provided either by simulation or monitoring. The simulation is proven as a very flexible technique that can be used to obtain an assessment of the current process performance and/or to formulate hypotheses about possible process redesign [11]. However cost and effort for setting up the simulation environment could sometimes prevent its use. Monitoring enables continuous collection of run-time data and analysis of QoS metrics that allow to discover the main factors that influence the process performance [12], [13].

Recently, monitoring approaches and facilities have been proposed such as: PROMO [2] which provides a KPI editor and facilities for monitoring and analysis of the BP; and ProM tool which analyzes BP constraints by means of temporal logic and coloured automata. Finally, the work in [3] presents an analytical comparison of KPI evaluation and monitoring solutions of business process and proposes a flexible, adaptable and dynamic monitoring infrastructure that is independent from any specific business process modeling notation and execution engine. In this paper, we leverage this monitoring solution to allow performance analysis and optimization of the business process. In addition, with respect to existing monitoring approaches for business process, our solution relies on usage of formal approaches [14], [15] enabling efficient analysis of BPMN models and the identification of performance criticalities in the BPMN specification.

III. Business Process Performance Analysis

The generic concept of BP execution adequacy criterion has been presented for the first time in [4], where adequacy has been connected with the identification of the relevant entities (Activity Entity, Sequence Flow Entity, Path Entity) to be covered during the BP execution.

In this paper, we leverage this concept of BP execution adequacy criteria so that they can be used to perform business process performance analysis. In this last case through a monitoring activity, the tasks or the set of tasks, that may cause bottleneck, delay or any other kind of performance violation can be identify and the BPMN model properly optimized. Without binding our proposal to a specific coverage adequacy criterion, we want to assess the performance by defining specific BP performance metrics, identifying what are the performance goals of the covered entities, and discovering the entities that are critical for the achievement of the target performance goal. As for test adequacy, if some entities are not covered during BP execution, we cannot exclude that these might hide some performance problems or security flaw. To do this, we use a proof-of-concept monitoring framework to observe and collect BP execution traces and measure the defined performance metrics considering the entities belonging to these traces. In the rest of this section, we define a set of BP performance metrics and the monitoring framework able to assess them.

A. Business Process Performance Metrics

The metrics we consider to perform BP performance analysis and optimization are:

a) Waiting Time: The waiting time is the time an activity is waiting for a resource to become available.

b) Synchronization Time: The synchronization time is the time an activity is not yet fully enabled and waiting for an external trigger or another parallel branch. Unlike waiting time, the activity is not fully enabled yet, i.e., the case is waiting for synchronization rather than a resource.

c) Path Execution Time: Considering *n* the number of tasks executed by the user during a business process execution, and t_i the time for the completion of the task *i*, the path execution time (*PC_time*) is computed as follows:

$$PC\_time = \sum_{i}^{n} t\_i$$

d) Path Coverage: The business process coverage (*bp_coverage*) is the number of different paths executed during a certain period over the total amount of paths in the BP. If k is the number of different path it is computed as:

$$bp\_coverage = \frac{k}{\#path}$$

e) Path Frequency: Given n the fixed number of observations of a BP the *pf_coverage* is computed as the number of executions of a specific path (k), over n as in the following:

$$pf\_coverage = \frac{k}{n}$$

B. Monitoring Framework

With reference to Figure 1, we present in this section the components of a monitoring framework able to measure the business process performance metrics defined in Section III-A:

- BPMN Path Explorer. This component is in charge to explore and save all the possible entities (Activity Entity, Sequence Flow Entity, Path Entity) reachable on a BPMN. The paths extraction is realized by coverability graph algorithm that exploits the advantages provided by the use of BPMN 2.0. It adopts a model transformation to translate the business process into a formal model according to the methodology presented in [14]. Specifically, it transforms (a subset of) BPMN models into Petri nets, then from Petri net a coverability graph is derived for the path exaction. Once extracted, the paths will be provided to the Rules Manager that through the Rules Generator will create, using the templates of rules stored into the Template Manager, a set of rules that aims to check the performance metrics defined on the business process.

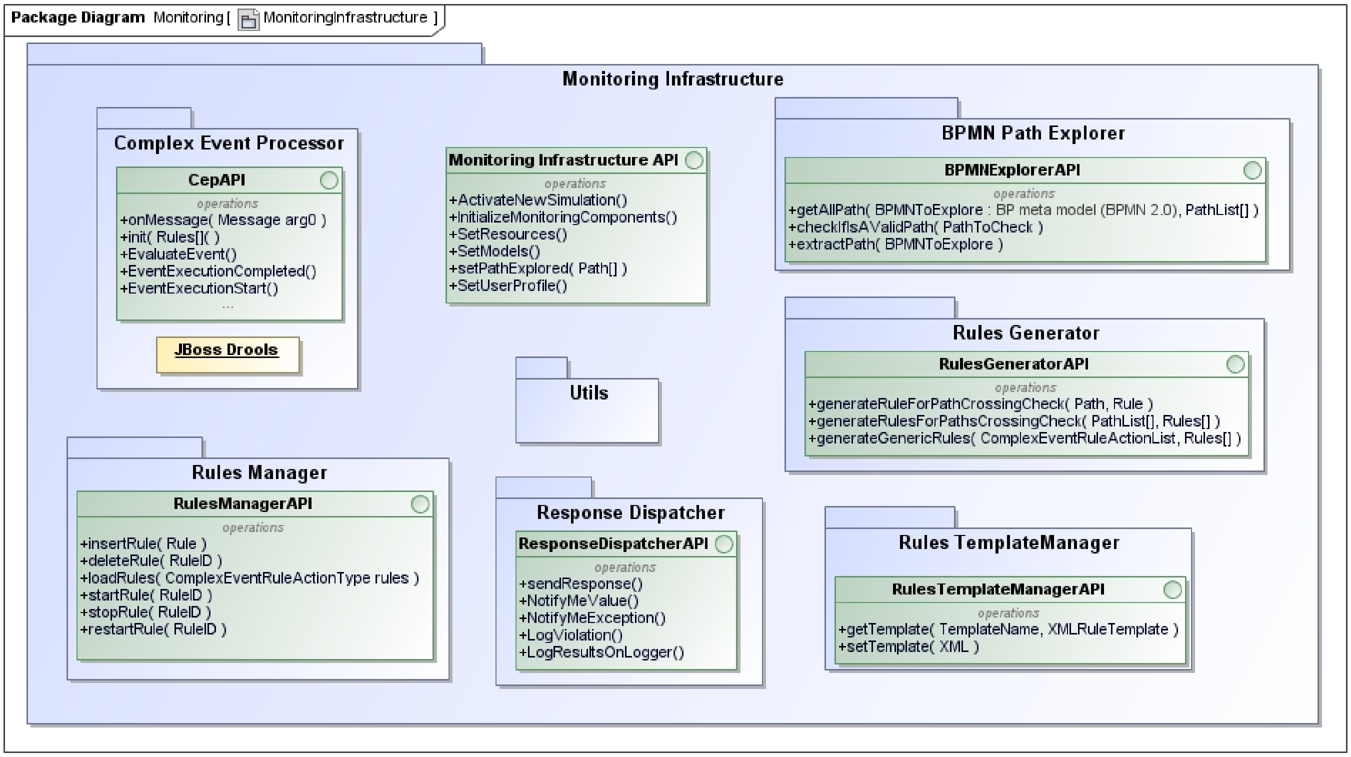

Fig. 1: Monitoring Framework

- Complex Event Processor (CEP). It is the rule engine which analyzes the events, generated by the business process execution. Several rule engines can be used for this task like Drools Fusion, VisiRule, RuleML. Our instance is realized using Drools Fusion [16], that is able to detect patterns and monitor the business process performance metrics.
- Rules Generator. The Rules Generator is the component in charge to generate the rules needed for the monitoring of the business process execution and the assessment of the performance metrics. It uses the templates stored into the Rules Template Manager. These rules are generated according to the specific performance metrics to be assessed. A generic rule consists of two main parts: in the first part the events to be matched are specified; the second part includes the events/actions to be notified after the rules evaluation.
- Rules Template Manager. This component is an archive of predetermined rules templates that will be instantiated by the Rules Generator. A rule template is a rule skeleton, the specification of which has to be completed by instantiating a set of template-dependent placeholders. The instantiation will refer to appropriate values inferred from the specific performance metrics to be assessed. Once the synthesis of the new set of rules is completed, the new rules are loaded by the Rule Generator into the Rules Template Manager.
- Rules Manager. The complex event detection process depends directly from the operation done by the Rules Manager component which is in charge to load and unload a set of rules into the complex event processor and fire it when needed.
- Response Dispatcher. The Response Dispatcher is a registry that records the requests of business process performance assessment. Once it receives the advice of a rule firing or pattern completion from the CEP, it stores the data related to the measured performance metrics. It elaborates statistics about the measured performance metrics and raises warning messages if the performance constraints defined on the business process are violated to the consumer/requester of the business process performance analysis. We refer to [3] for a detailed description of the monitoring framework.

IV. Case Study

In this section we present an example of use of the proposed approach considering a case study representing a food delivery service (FDS), which is able to provide either pizza or primo to its customers. The claim of the food delivery service is that each meal will be delivered in 30 minutes, otherwise a gift (an extra drink, cookies, or voucher) will be added to the order. The target is to deliver each order in time so to decrease the number of gifts and improve the food delivery service profits. In this paper, a simplified version of the Business Process Model (BPM) used by the FDS is presented as described in the following section.

A. Business Process Description

In the case study considered, the BPM involves five different stakeholders that are (see Figure 2): the *customer*, the *Clerk*, the *Pizza Chief*, the *Restaurant Chief* and *Delivery Boy*. In the BPM, the *customer* makes an order which is received by the *Clerk* and forwarded to either the *Pizza Chief* or the *Restaurant Chief* in order to prepare the proper meal (represented by the task *Bake Pizza* (BP) and *Make Primo* (MP) respectively). Once the order is completed, i.e. task *Order Received* (OR), the *Delivery Boy* delivers the prepared meal to the customer, i.e. task *Delivery Order* (DO). The BPM manages also the case in which the pizza or primo preparation is starting to go out of time (i.e. over 9 minutes for pizza and over 12 for primo) so that a gift has to be added for free to the customer order, i.e. task *Add Gift* (AG). Finally, the *Delivery Boy* delivers the prepared meal to the customer and receives the payment, i.e. task *Receive Payment* (RP). Only in case this last task terminates with an overall completion over the 30 minutes the gift is actually delivered to the customer, i.e. task *Delivery Gift* (DG).

B. Experimental Setup

The evaluation of the proposed approach has been performed with the collaboration of a class of 16 students of a Software Engineering Course. The students were randomly divided into 4 groups, each one playing a specific business process scenario relative to the model described in the previous section. In particular, students know that estimated task completion time required to *OR*, *BP*, *MP*, *DO* and *RP* are $1'30''$, $6'$, $8'$ $10'$ and $2'$ respectively. The BMP has to be able to manage a 50% deviation of the estimated task completion time. At the end, data collected has been analyzed by the students and optimization suggestion proposed. To make the results analysis unbiased, each students group did not analyze the data of the own experiment. For aim of simplicity the measures for waiting time and the synchronization time have not been considered in the experiments and a single deviation has been introduced in each experiment.

The scenarios considered are:

1) *standard execution*: Each task of the BPM satisfy the 50% deviation of the estimated completion time. In this case the student uses the following time range as: $1'50''$ - $2'2''$ for *RO*, $6'$ - $9'$ for *BP*, $8'$ - $12'$ for *MP*, $10'$ - $15'$ for *DO* and $2'$ - $3'$ for *RP*. We leave the students to select pizza and primo according to their preferences. In this case the results should confirm that the overall business process completion time is always within $30'$ and no gift is delivered.
2) *delay in pizza or primo preparation*: Both the tasks *BP* and *MP* have 100% deviation of the estimated task completion time. In this case the student uses the following modified time range: $6'$ - $12'$ for *BP*, $8'$ - $16'$ for *MP*. For a simpler analysis students equally select pizza or primo during the scenario execution. The final analysis of the results should: let the individuation of the business process bottlenecks; estimate deviation from the overall business process completion time ($30'$); provide estimation of the number of gifts delivered and therefore loss in profit; suggest possible BPM optimizations.
3) *delay in meal delivery*: The task *DO* has 100% deviation of the estimated task completion time therefore its time range becomes $10'$ - $20'$. As in the *delay in pizza or primo preparation* scenario students equally select pizza or primo during execution and results of the final analysis should be the same of that experiment.
4) *delay in receive payment*: The task *RP* has 100% deviation of the estimated task completion time therefore its time range becomes $2'$ - $4'$. In this case the service implementing the task *RP* is provided by an external service provider. As in the *delay in pizza or primo preparation* scenario, students equally select pizza or primo during execution and results of the final analysis should be the same of that experiment.

To speed up the simulation of the different scenarios and let the students to complete the experimentations in a lecture duration (1h30m) we reduced the time in proportion so that an execution was no longer than $10'$. Consequently each students group performed 12 executions of the BPM. The BPM execution has been performed using an implementation of the described framework relying on the web interface of Activiti [17] engine. Specifically, a user for each role described on the BPMN has been created on the execution platform and assigned to a student apart from the roles of *Clerk* and *Customer* that were executed from the same student. Through the data collected during the execution and computed by using the Complex Event Processor component of the monitoring framework, the performance metrics defined in Section III-A are calculated and stored into an internal database for the subsequent students analysis.

C. First results collection

The results collected by each group are shown in the tables below. In particular, Table I presents the data related to the *standard execution*. The first two rows represent the name of the tasks and the admissible task completion ranges. In the remaining rows there are the task completion times collected during the different executions. In particular, the columns labeled RO (*Receive Order*), BP (*Bake Pizza*), MP (*Make Primo*), DO (*Delivery Order*), RP (*Receive Payment*) report two values: the computed task completion time and a *Y/N* label in case the task completion time is over/below the admissible 50% deviation. The column labeled AG (*Add gift*) is set to *Y* if the time for preparing pizza or primo is starting to go out of time (i.e. over $9'$ for pizza and over $12'$ for primo), *N* otherwise. The column labeled, GD (*Gift Delivered*) is set to *Y* if the gift is actually given to the client, i.e. the overall completion of the business process is greater than $30'$, *N* otherwise. Finally, the column labeled *PC_Time* reports the overall completion time for delivering the food to the customer. Likewise Table II, Table III and Table IV show the

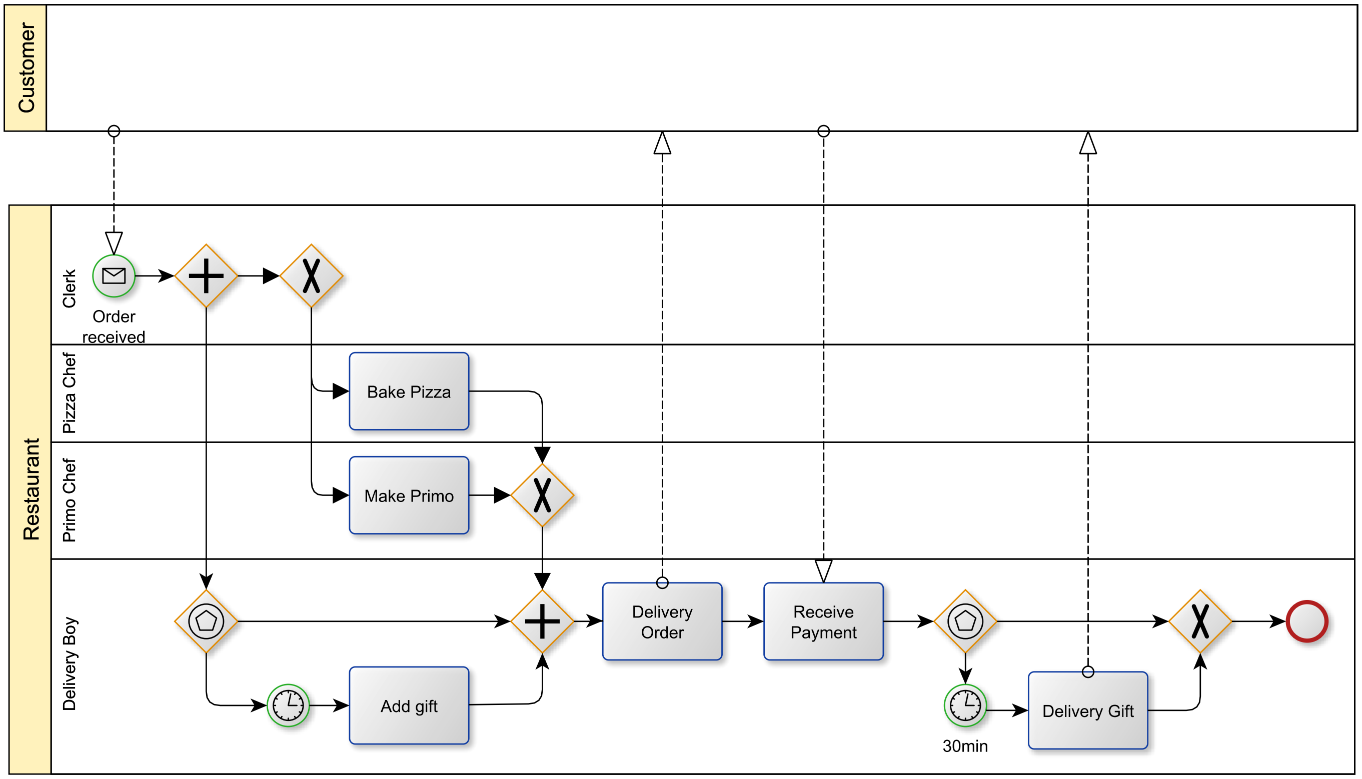

Fig. 2: The BPMN Main Process

executions of the remaining scenarios: delay in pizza or primo preparation, delay in meal delivery, delay in receive payment respectively. In all the tables, the last row collects the values related to *Path Coverage* and *Path Frequency*. In this case for the BPM of Figure 2 the path cardinality is 6 as evidenced in the following:

- P1:*BP*→*DO*→*RP*,
- P2:*MP*→*DO*→*RP*,
- P3:*BP*→*AG*→*DO*→*RP*,
- P4:*MP*→*AG*→*DO*→*RP*,
- P5:*BP*→*AG*→*DO*→*RP*→*DG*,
- P6:*MP*→*AG*→*DO*→*RP*→*DG*.

Considering Table I and the performance measures collected during the executions is possible to observe: i) for each execution the path execution time (*PC_time*) is less than $30'$; ii) the *Path Coverage* is equal to 2/6, because the paths involving the *Add Gift* task are never executed and the column labeled, *Gift Delivered* is always set to *N*. This confirm that the overall business process execution time is always within $30'$ as required by the FDS specification and no loss in profit is experienced because no gift is delivered. However, the analysis of *Path Frequency* reveals that customers (students in this case) like more pizza than primo. Path involving a pizza order has (8/12) as path frequency. This could be an important information in case the FDS owner would like to improve its benefits and invest more on the most likely meal preparation.

Considering Table II, for 3 of the 12 executions the *PC_time* is over than $30'$ evidencing that there are problems in the FDS performance. A detailed analysis of the tasks completion time of these executions reveals that the responsible are *BP* and *MP* (as in the scenario intentions). Because each gift delivered means a decreasing in the profit for the FDP, the collected data suggested to the FDS to improve the performance of the bottleneck activities. This could be implemented for instance employing more personnel or speeding up the cooking time. However, the table reveals also some other particular situations: for instance in experiments 9 and 10 the task *MP* is in delay however the gift has not been delivered to the customer. Indeed there is *Y* in column labeled *AG* and *N* in column labeled *GD*. As evidenced by the table, the gift delivery decision strictly depends on the sum of completion time of the tasks *DO* and *RP*. Therefore, reducing the *DO* completion time could make the FDS business process tolerant to deviation greater than 50%. Finally, the *Path Coverage* value reveals that the 100% coverage is never reached. Then, even if the *BP* task has a 100% delay, the completion time of the remaining tasks avoids the delivery of a gift. This could be an important information for the FDS because it could decide redistribute some of its personnel preparing the pizza on the *Make Primo* task to speed up the overall completion time and avoid loss in profit. As evidenced by this simple scenario, the data and the measures provided by the monitoring activity during the BPM execution, let the individuation of the business process bottlenecks and provide estimation of the overall business process completion time useful for cost estimation, providing suggestion for possible BPM optimizations.

Considering Table III, for 2 of the 12 executions the *PC_time* is over than $30'$. Collected data evidence that there are problems in the task *DO*. The frequency of such a situation is not very annoying and FDS could decide to live the BPM as it is. However, looking at the column labeled *GD* it is possible to notice some very dangerous situations: the completion time is over $30'$ and no gift is delivered to the customer. See for instance Experiment 11 and 12. This is against the specification of FDS and could be very bad mark for its repudiation. A modification of the BPM is therefore necessary. Solutions could include the employment of more personnel for the delivery or a structural modification of the BPMN itself so to always provide the delivery boy of a gift to be used in case of delay. Thanks to the monitoring data collected during the executions of the business process, bottleneck has been identified and an important flaw of the BPM discovered.

Considering Table IV, for just 1 over the 12 executions the *PC_time* is over than $30'$ due to the delay of *RP*. As in the previous scenarios, the frequency of such a situation is not very annoying and FDS could decide to live the BPM as it is. However, again the same dangerous situation can be evidenced. Differently from the previous scenario, in this case the service implementing the *RP* task here is implemented by an external service provider therefore FDP has no control on the service performance. Unique solution could be providing the delivery boy with a gift to be used in case of delay, or change the service provider with one having better performance. Again thanks to the monitoring data collected during the executions of the business process, bottleneck has been identified and possible solutions identified.

TABLE I: Simulation of standard execution

	RO	**BP**	**MP**	**AG**	**DO**	**RP**	**GD**	**PC_Time**
Range:	*1,5m-2,2m*	*6m-9m*	*8m-12m*		*10m-15m*	*2m-3m*		
	Xp/Del	*Xp/Del*	*Xp/Del*		*Xp/Del*	*Xp/Del*		
Ex.1	2/N	6/N		N	10/N	3/N	N	19
Ex.2	2/N	9/N		N	14/N	3/N	N	26
Ex.3	2/N	7/N		N	14/N	2/N	N	22
Ex.4	2/N	7/N		N	15/N	2/N	N	24
Ex.5	2/N	6/N		N	13/N	3/N	N	22
Ex.6	2/N	6/N		N	15/N	2/N	N	23
Ex.7	2/N	8/N		N	14/N	3/N	N	25
Ex.8	2/N	7/N		N	10/N	3/N	N	20
Ex.9	2/N		9/N	N	14/N	3/N	N	26
Ex.10	2/N		9/N	N	12/N	2/N	N	23
Ex.11	2/N		9/N	N	15/N	2/N	N	26
Ex.12	2/N		8/N	N	11/N	2/N	N	21
Path Frequency: P1:8/12; P2:4/12; P3:0; P4:0; P5:0; P6:0 **Path Coverage:** 2/6								

TABLE II: Simulation of delay in primo/pizza preparation

	RO	**BP**	**MP**	**AG**	**DO**	**RP**	**GD**	**PC_Time**
Range:	*1,5m-2,2m*	*6m-12m*	*8m-16m*		*10m-15m*	*2m-3m*		
	Xp/Del	*Xp/Del*	*Xp/Del*		*Xp/Del*	*Xp/Del*		
Ex.1	2/N	6/N		N	15/N	2/N	N	23
Ex.2	2/N	**10/Y**		**Y**	11/N	3/N	N	24
Ex.3	2/N	**12/Y**		**Y**	11/N	2/N	N	25
Ex.4	2/N	7/N		N	11/N	3/N	N	21
Ex.5	2/N	7/N		N	10/N	2/N	N	19
Ex.6	2/N	9/N		N	12/N	2/N	N	23
Ex.7	2/N		**16/Y**	**Y**	15/N	2/N	**Y**	**33**
Ex.8	2/N		**16/Y**	**Y**	12/N	3/N	**Y**	**31**
Ex.9	2/N		**16/Y**	**Y**	11/N	2/N	N	29
Ex.10	2/N		**13/Y**	**Y**	12/N	2/N	N	27
Ex.11	2/N		12	N	14/N	3/N	N	29
Ex.12	2/N		**16/Y**	**Y**	15/N	3/N	**Y**	**34**
Path Frequency: P1:4/12; P2:1/12; P3:2/12; P4:2/12; P5:0; P6:3/12 **Path Coverage:** 5/6								

TABLE III: Simulation of delay in meal delivery

	RO	**BP**	**MP**	**AG**	**DO**	**RP**	**GD**	**PC_Time**
Range:	*1,5m-2,2m*	*6m-9m*	*8m-12m*		*10m-20m*	*2m-3m*		
	Xp/Del	*Xp/Del*	*Xp/Del*		*Xp/Del*	*Xp/Del*		
Ex.1	2/N	9/N		N	**16/Y**	3/N	N	28
Ex.2	2/N	8/N		N	10/N	2/N	N	20
Ex.3	2/N	8/N		N	14/N	3/N	N	25
Ex.4	2/N	7/N		N	**20/Y**	3/N	N	30
Ex.5	2/N	7/N		N	**16/Y**	2/N	N	25
Ex.6	2/N	7/N		N	11/N	3/N	N	21
Ex.7	2/N		10/N	N	15/N	2/N	N	27
Ex.8	2/N		8/N	N	10/N	2/N	N	20
Ex.9	2/N		8/N	N	15/N	2/N	N	25
Ex.10	2/N		10/N	N	12/N	2/N	N	24
Ex.11	2/N		11/N	N	**17/Y**	3/N	N	**31**
Ex.12	2/N		12/N	N	**18/Y**	2/N	N	**32**
Path Frequency: P1:3/12; P2:4/12; P3:3/12; P4:2/12; P5:0; P6:0 **Path Coverage:** 4/6								

TABLE IV: Simulation of delay in receive payment

	RO	**BP**	**MP**	**AG**	**DO**	**RP**	**GD**	**PC_Time**
Range:	*1,5m-2,2m*	*6m-9m*	*8m-12m*		*10m-15m*	*2m-4m*		
	Xp/Del	*Xp/Del*	*Xp/Del*		*Xp/Del*	*Xp/Del*		
Ex.1	2/N	9/N		N	13/N	**3/N**	N	25
Ex.2	2/N	8/N		N	15/N	3/N	N	26
Ex.3	2/N	7/N		N	11/N	**4/Y**	N	22
Ex.4	2/N	9/N		N	15/N	3/N	N	27
Ex.5	2/N	8/N		N	14/N	**4/Y**	N	26
Ex.6	2/N	7/N		N	13/N	3/N	N	23
Ex.7	2/N		10/N	N	13/N	3/N	N	26
Ex.8	2/N		11/N	N	10/N	3/N	N	24
Ex.9	2/N		9/N	N	10/N	3/N	N	22
Ex.10	2/N		9/N	N	13/N	**4/Y**	N	26
Ex.11	2/N		12/N	N	15/N	**4/Y**	N	**31**
Ex.12	2/N		11/N	N	11/N	**4/Y**	N	26
Path Frequency: P1:6/12; P2:6/12; P3:0; P4:0; P5:0; P6:0 **Path Coverage:** 2/6								

D. Business Process Optimization

In this section, we present an optimized version of the BPMN Main Process as in Figure 3 and the results of a new experiment considering this improved version of the business process.

As evidenced by the experimental results presented in Section IV-C, one of the main bottlenecks in the business process that lets to overcome the overall completion time and produces a loss in profit is the *Make Primo* task.

The main improvement of the BPMN Main Process presented in Figure 3 with respect to the original BPMN presented in Figure 2 is that two stakeholders (*Chief Primo1* and *Chief Primo2*) are in charge to prepare in parallel the *Primo* meal (represented by the tasks *Make Primo1* and *Make Primo2*).

We asked a class of 16 students of the Software Engineering Course to perform a new experiment with the improved version of the business process. The students were randomly divided in two groups, each one playing a specific business process scenario. In particular, the scenarios considered in this new experiment were:

1) *standard execution*: In this case the student uses the following time ranges: $1'50''$ - $2'20''$ for *RO*, $6'$ - $9'$ for *BP*, $8'$ - $10'$ for *MP*, $10'$ - $15'$ for *DO* and $2'$ - $3'$ for *RP*. Note that with respect to the time ranges of the previous experiment, we decreased the time of *Make Primo* since this task is executed in parallel by two stakeholders in the improved version of the business process. We leave the students to select pizza and primo according to their preferences. As in the *standard execution* scenario of the experiment presented in Section IV-C, the results should confirm that the overall business process completion time is always within $30'$ and no gift is delivered.
2) *delay in primo preparation*: In this case the student uses the following time ranges: $6'$ - $9'$ for *BP*, $8'$ - $14'$ for *MP*. For a simpler analysis students equally select pizza or primo during the scenario execution. The task *MP* has 75% deviation of the estimated task completion time (this is lower than the deviation of the estimated task completion time (100%) considered in the experiment presented in Section IV-C since this task is executed in parallel by two stakeholders in the improved version of the business process).

TABLE V: Simulation of standard execution with the improved BP

	RO	BP	MP	AG	DO	RP	GD	PC_Time
Range:	*1,5m-2,2m*	*6m-9m*	*8m-10m*		*10m-15m*	*2m-3m*		
	Xp/Del	*Xp/Del*	*Xp/Del*		*Xp/Del*	*Xp/Del*		
Ex.1	2/N	8/N		N	13/N	3/N	N	24
Ex.2	2/N	9/N		N	11/N	3/N	N	23
Ex.3	2/N	8/N		N	14/N	3/N	N	25
Ex.4	2/N	8/N		N	14/N	3/N	N	25
Ex.5	2/N	9/N		N	11/N	2/N	N	22
Ex.6	2/N	8/N		N	11/N	3/N	N	22
Ex.7	2/N	6/N		N	12/N	3/N	N	21
Ex.8	2/N	8/N		N	14/N	3/N	N	25
Ex.9	2/N		8/N	N	14/N	3/N	N	25
Ex.10	2/N		9/N	N	11/N	3/N	N	23
Ex.11	2/N		8/N	N	14/N	3/N	N	25
Ex.12	2/N		9/N	N	13/N	2/N	N	24
Path Frequency: P1:8/12; P2:4/12; P3:0; P4:0; P5:0; P6:0 **Path Coverage:** 2/6								

Table V shows the results for the *standard execution* scenario with the improved business process whereas Table VI shows the results for the *delay in primo preparation* scenario. As in the previous experiment, the task completion times are presented and *Y/N* label indicates that the task completion time is over/below the admissible 50% deviation. As in the previous experiment, the column labeled AG (*Add gift*) is set to *Y* if the time for preparing pizza or primo is starting to go out of time (i.e. over $9'$ for pizza and over $12'$ for primo), *N* otherwise. The column labeled, GD (*Gift Delivered*) is set to *Y* if the gift is actually given to the client, i.e. the overall completion of the business process is greater than $30'$, *N* otherwise. The same performance measures of the previous experiment are collected. In particular, the results presented in Table V, as in the previous experiment, show that : i) each path execution time (*PC_time*) is less than $30'$; ii) the *Path Coverage* is equal to 2/6, because the paths involving the *Add Gift* task are never executed and the column labeled *Gift Delivered* is always set to *N*. This evidences that in the standard execution scenario, the optimization introduced in the business process model does not affect business process performance.

TABLE VI: Simulation of delay in primo preparation with the improved BP

	RO	BP	MP	AG	DO	RP	GD	PC_Time
Range:	*1,5m-2,2m*	*6m-9m*	*8m-14m*		*10m-15m*	*2m-3m*		
	Xp/Del	*Xp/Del*	*Xp/Del*		*Xp/Del*	*Xp/Del*		
Ex.1	2/N	9/N		N	12/N	3/N	N	24
Ex.2	2/N	6/N		N	12/N	2/N	N	20
Ex.3	2/N	8/N		N	14/N	2/N	N	24
Ex.4	2/N	9/N		N	10/N	3/N	N	22
Ex.5	2/N	7/N		N	10/N	3/N	N	20
Ex.6	2/N	9/N		N	10/N	3/N	N	22
Ex.7	2/N		**14/Y**	**Y**	15/N	3/N	**Y**	**31**
Ex.8	2/N		**10/N**	**N**	11/N	2/N	**N**	24
Ex.9	2/N		**10/N**	**N**	15/N	3/N	N	27
Ex.10	2/N		**11/N**	**N**	12/N	3/N	N	26
Ex.11	2/N		12/N	N	12/N	2/N	N	26
Ex.12	2/N		**13/Y**	**Y**	12/N	3/N	**N**	28
Path Frequency: P1:6/12; P2:4/12; P3:0; P4:1/12; P5:0; P6:1/12 **Path Coverage:** 4/6								

Table VI shows that in case of *delay in primo preparation* scenario, there is an improvement of the business process performance. In particular, this table reveals that: i) the number of added gifts as well as the number of delivered gifts decreases of more than 50% with respect to the previous experiment; ii) there is one execution (with respect to the 3 executions in the previous experiment) having a *PC_time* over than $30'$; iii) for 2 executions (with respect to 5 of the previous experiment) the task *MP* is in delay, however the gift is delivered to the customer only for one execution (Ex.7) since the gift delivery decision strictly depends on the sum of completion time of the tasks *DO* and *RP*; iv) in the improved version of the business process there are only three different combinations of the completion times values of the tasks *MP*, *DO* and *RP* allowing for an overall completion time of the business process greater than $30'$. For the scenario of *delay in primo/pizza preparation* in the original version of the business process, this combinations number is equal to 16; v) there is the same path cardinality as for the original business process model whereas the *Path Coverage* is lower $(4/6)$ with the respect of $5/6$ of Table II. This is due not to the business process improvement but to the different *Path Frequency*.

Moreover, similar optimizations to that provided for *Make Primo* task can be applied for the other business process tasks such as *Bake Pizza* and *Delivery Order*. We expect that these business process optimizations will further improve the business process performance in terms of overall path

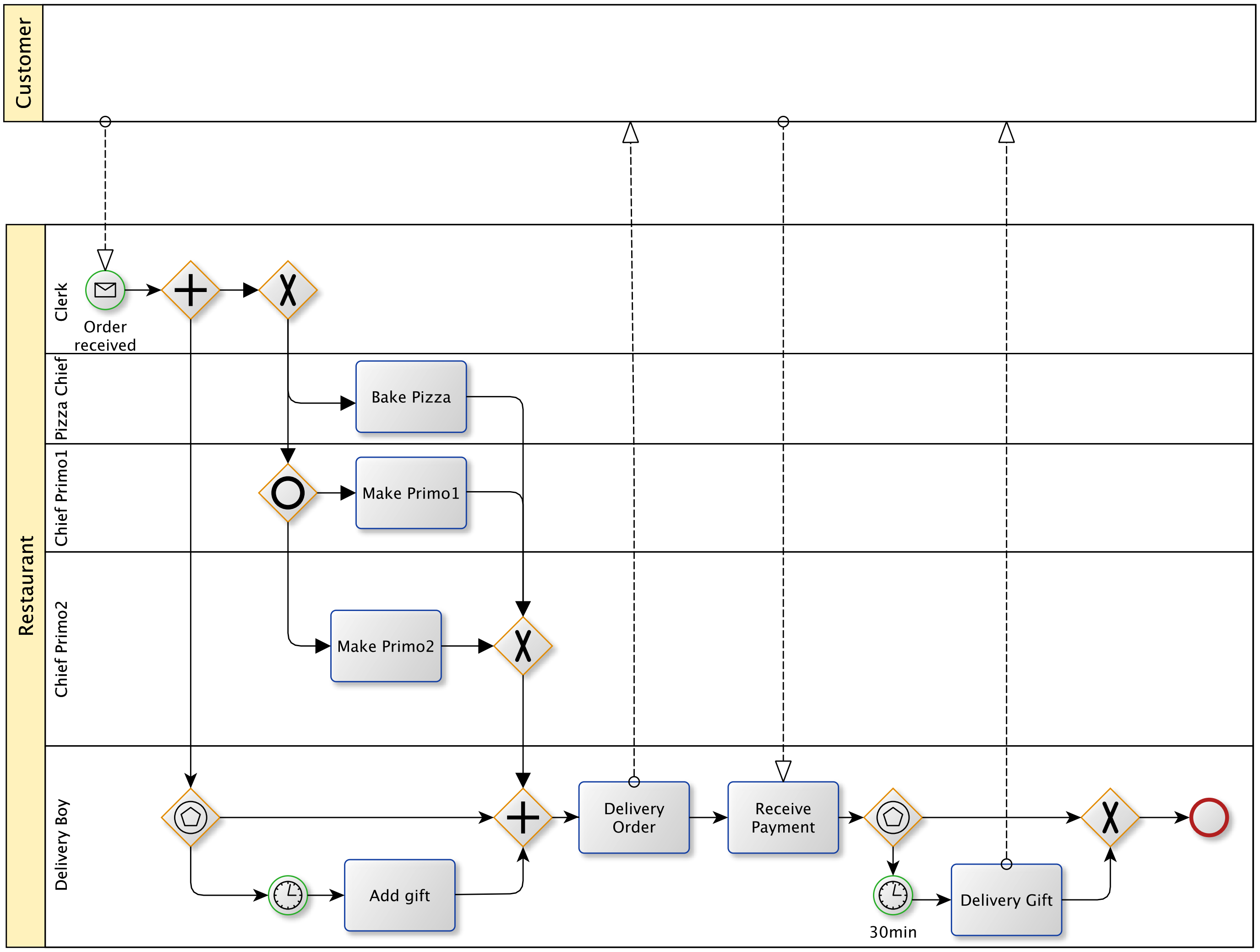

Fig. 3: The improved BPMN Main Process

completion time and the numbers of delivered gifts.

V. Conclusion

Business process performance analysis is a key factor for supporting business process management and improvement of current process practices. Monitoring facilities are commonly used for KPIs measurement and business process performance assessment. In this paper, we define specific business process performance metrics leveraging coverage adequacy criteria and rely on a flexible and adaptable event based monitoring framework for assessment of the defined metrics. The proposed approach is able to monitor business process executions and discover the entities that are critical for the achievement of the overall performance of the business process.

We presented an example of use of the proposed approach considering a case study representing a food delivery service. This case study showed the effectiveness of the proposed framework in measuring the defined performance metrics, such as *path coverage*, *path frequency*, *path execution time*.

Moreover, during a first validation the obtained results for these metrics revealed bottleneck activities as well as violations of performance constraints in the business process, suggesting improvements and optimizations of the process itself. We presented a new version of the business process addressing some of the suggested improvements, and showed that the provided business process improvement revealed a performance optimization of the business process in terms of overall path completion time and delivered gifts. We plan to evaluate the effects of further optimizations of the same business process.

In future work, we intend to refine and enhance the proposed performance metrics in order to exploit historical data related to consecutive business process executions. A further research direction deals with investigating automated optimization facilities for implementing possible process improvements.

Acknowledgment

This work has been partially funded by the Model-Based Social Learning for Public Administrations project (EU FP7-ICT-2013-11/619583).

References

[1] B. Wetzstein, P. Leitner, F. Rosenberg, I. Brandic, S. Dustdar, and F. Leymann, "Monitoring and analyzing influential factors of business process performance," in *Enterprise Distributed Object Computing Conference*, Sept 2009, pp. 141–150.

[2] P. Bertoli, M. Dragoni, C. Ghidini, E. Martufi, M. Nori, M. Pistore, and C. Di Francescomarino, "Modeling and monitoring business process execution," in *Service-Oriented Computing*. Springer, 2013, pp. 683–687.

[3] A. Calabrò, F. Lonetti, and E. Marchetti, "Monitoring of business process execution based on performance indicators," in *41st Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, 2015, pp. 255–258.

[4] A. Bertolino, A. Calabrò, F. Lonetti, and E. Marchetti, "Towards business process execution adequacy criteria," in *Software Quality. The Future of Systems-and Software Development*. Springer, 2016, pp. 37–48.

[5] K. H. Han, J. G. Kang, and M. Song, "Two-stage process analysis using the process-based performance measurement framework and business process simulation," *Expert Systems with Applications*, vol. 36, no. 3, pp. 7080–7086, 2009.

[6] X. Meng and M. Minogue, "Performance measurement models in facility management: a comparative study," *Facilities*, vol. 29, no. 11/12, pp. 472–484, 2011.

[7] H. A. Bassioni, A. D. Price, and T. M. Hassan, "Building a conceptual framework for measuring business performance in construction: an empirical evaluation," *Construction Management and Economics*, vol. 23, no. 5, pp. 495–507, 2005.

[8] D. Amaratunga, D. Baldry, and M. Sarshar, "Process improvement through performance measurement: the balanced scorecard methodology," *Work study*, vol. 50, no. 5, pp. 179–189, 2001.

[9] Z. Jin, F. Deng, H. Li, and M. Skitmore, "Practical framework for measuring performance of international construction firms," *Journal of Construction Engineering and Management*, vol. 139, no. 9, pp. 1154–1167, 2013.

[10] K. Vergidis, A. Tiwari, and B. Majeed, "Business process analysis and optimization: Beyond reengineering," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 38, no. 1, pp. 69–82, 2008.

[11] W. M. Van der Aalst, J. Nakatumba, A. Rozinat, and N. Russell, "Business process simulation," in *Handbook on Business Process Management 1*. Springer, 2010, pp. 313–338.

[12] F. Koetter and M. Kochanowski, "A model-driven approach for event-based business process monitoring," *Information Systems and e-Business Management*, vol. 13, no. 1, pp. 5–36, 2015.

[13] B. Wetzstein, P. Leitner, F. Rosenberg, I. Brandic, S. Dustdar, and F. Leymann, "Monitoring and analyzing influential factors of business process performance," in *IEEE International Enterprise Distributed Object Computing Conference (EDOC'09)*, 2009, pp. 141–150.

[14] R. M. Dijkman, M. Dumas, and C. Ouyang, "*Semantics and analysis of business process models in BPMN*," *Information and Software Technology*, vol. 50, no. 12, pp. 1281–1294, 2008.

[15] E. Verbeek and W. M. Van Der Aalst, "Woflan 2.0 a petri-net-based workflow diagnosis tool," in *Application and Theory of Petri Nets 2000*. Springer, 2000, pp. 475–484.

[16] "Drools Fusion: Complex Event Processor," http://www.jboss.org/drools/drools-fusion.html.

[17] "Activiti BPM Platform," http://activiti.org/.