# Leveraging Regression Algorithms for Predicting Process Performance Using Goal Alignments

Karthikeyan Ponnalagu[1(✉)], Aditya Ghose[2], and Hoa Khanh Dam[2]

[1] Robert Bosch, Bangalore, India
karthikeyan.ponnalagu@in.bosch.com
[2] University of Wollongong, Wollongong, Australia
aditya.ghose@gmail.com, hoa@uow.edu.au

**Abstract.** Industry-scale context-aware processes typically manifest a large number of variants during their execution. Being able to predict the performance of a partially executed process instance (in terms of cost, time or customer satisfaction) can be particularly useful. Such predictions can help in permitting interventions to improve matters for instances that appear likely to perform poorly. This paper proposes an approach for leveraging the process context, process state, and process goals to obtain such predictions.

**Keywords:** Variability · Contextual factor analysis · Business process mining

## 1 Introduction

Execution of complex business processes that are specifically knowledge driven, generally leads to significant amounts of event records corresponding to the execution of activities in the processes. Most of the current literature assumes that the performance of a process instance is entirely determined by what happens over the course of the execution of the process instance. We see limitations in such assumptions [7], when applied in knowledge intense process models, where the specific instance executions are dictated by other factors that are not part of process executions. In this paper, we propose a novel approach that inter-operates with cloud based cognitive systems towards predicting process performance. Towards this, we leverage contextual factors and goal alignments associated with the actual execution of processes.

We assume the following inputs to our proposed approach: (a) a goal model hyper graph with goals and sub-goals (AND, OR) represented as a collection of boolean conditions in conjunctive normal form (CNF), (b) an event log containing multiple process instance execution data and (c) a process design annotated with normative end effects. In this paper, we consider an `Incident management` process design as our running example. A process log[1] containing 1400 executed

[1] https://www.scribd.com/document/333254045/IncidentLog.


F. Daniel et al. (Eds.): BPM 2018 Workshops, LNBIP 342, pp. 325–331, 2019.
https://doi.org/10.1007/978-3-030-11641-5_26

instances of this process design is considered for the evaluation of our proposed approach. A total of over 25000 task execution records is available as part of this process log. Each process instance in this log indicates how after receiving a complaint from a customer, an incident ticket is created, resolved and closed. We leverage annotated goal models with end effects. Such a goal model can be constructed through a goal refinement machinery as discussed in [2].

A variety of outcome predicting process monitoring techniques have been proposed in the literature [6]. In [4], the authors clearly establish the need for a general framework for mining and correlating business process characteristics from event logs. In [1], the authors discuss construction of a configurable process model as a family of process variants discovered from a collection of event logs. The existing works in the area of contextual correlation of business processes have addressed different challenges related to collaboration, contract conformance, process flexibility [5]. In comparison our work uses contextual factors and semantic effect traces on both partial and completed executions to correlate and predict execution deviation based on goal alignments. Works such as [3] focuses on generating performance predictions leveraging process simulation data. Works such as [9] focus on generating hybrid process model creation by leveraging event log clusters. In comparison, we focus on an orthogonal approach of discovering multiple process designs that are goal aligned variants of the original process design.

## 2 Identifying Process Context, Goals and Process State

Contextual information can be traced from process instances to a range of time-stamped information sources, such as statements being made on enterprise social media, financial market data, weather data and so on. Process log time-stamps can be correlated with time-stamps in these repositories of information to derive a wealth of information about the context within which a process instance was executed. In our proposed approach, we leverage this specific category of contextual information.

The performance indicators associated with process effect assertions are typically influenced with the entailment to specific OR-refinement sub goals (Email confirmation or Telephonic confirmation with customer) in the goal model. Given a state $S$ and a set of effect assertions $e$ obtained from events accruing from the execution of a task, the resulting partial state is given by $S \oplus e$, where $\oplus$ is a state update operator [8]. Similarly, given a normative state $S_N$ and a set of effect assertions $e_N$ obtained from events accruing from the execution of a task in a process, the resulting partial state is given by $S_N \oplus e_N$ where $\oplus$ is a state update operator. We also use a knowledge-base $KB$ of domain constraints. If $S \cup e \cup KB$ is consistent, then $S \oplus e = S \cup e$. Otherwise, $S \oplus e = e \cup \{s \mid s \subseteq S, s \cup e \cup KB$ is consistent, and there does not exist any $s'$ where $s \subset s' \subseteq S$ such that $s' \cup e \cup KB$ is consistent$\}$. We start with an initial partial state description (which may potentially be empty) and incrementally update it (using $\oplus$) until we reach the partial state immediately following the final task in the process instance. Towards achieving this, the proposed machinery leverage the OR-refinement goal correlations associated with each state transition from the process event log. For generating goal correlations based on the end effects (at the process or task levels), we have leveraged the `Process Instance Goal Alignment Model (PIGA)` discussed in our previous work [8]. Therefore, given a *goal-realizing effect group* S, finding correlation with a goal G in formal terms is simply finding the truth assignments in the CNF expression of G using the cumulative end effects of S. Towards generating PIGA, the list of state transitions and the goal decomposition model as input are considered. Then, for each event group in the process log, the truth assignments of all goals in the goal model are validated. This is repeated for all event groups in the process log to identify the "valid process instances". The representation of each process instance as a list of *maximally refined correlated goals* constitutes the completion of generating Process Instance Goal Alignment (PIGA).

**Table 1.** Context Correlated Goal Models (CCGM)

| No. of instances | Observed state effects | OR-refined goal entitlement | Context name (Value) |
|---|---|---|---|
| 62 | T4: (Resolution_Suggested) | (Link to Existing Problem, Close Problem) | CM1 = Connection('Remote', 'NotAvailable', 'BehindFirewall'), CustomerExpertise('High'), CustomerPriority('Low') |
| 155 | T3: (Resolution_Known) | (Link to Existing Problem, Close Problem) | CM2 = Solution('Known', 'AutoFix', 'BroadCast'), CustomerAffected('Group') |
| 11 | T5: (Resolution_Cancelled) | (Close Problem) | CM3 = Agent('New'), ProblemOrigin('3rd Party', 'NotUnderContract') |
| 51 | T5: (Ticket_NotEnriched) | (Escalate Problem) | CM4 = CustomerProvided ('NoEventTrace', 'NotReproduced') |
| 10 | T1: (Problem_NotCategorized), T9: (Problem_DetailIncomplete) | (Escalate Problem, Link to Existing Problem) | CM5 = Agent('New'), ProblemAutoCategory('Failed') |
| 5 | T2: (Problem_SeverityWarning), T3: (Set_TicketPriorityHigh) | (Escalate Problem, Enrich Problem) | CM6 = Agent('Expert') , ProblemAutoCategory('Complex') |
| 31 | T4: (Customer_NotNotified) | (Escalate Problem, Enrich Problem) | CM7 = CustomerSupport('Rare'), CustomerProvided('NoEventTrace', 'NotReproduced') |

The CCGM generated for our running example is illustrated in Table 1. For example as observed in row 3, 11 process instances are partially executed without a resolution to a reported incident due to a collection of contextual factors (CM3). To support predictions both at the process and individual task levels, we have leveraged two categories of effect log data sets: **Process Data Set(PD)**, where record in this data set is a tuple { Process Instance Identifier, a semantic trace, process execution time, context, aligned OR-refinement sub-goals } and **Task Data Set(TD)**: Each record in this data set is a tuple { Process Instance Identifier, Task Identifier, semantic trace from the execution of task, task execution time, total process execution time, context, task aligned goals, process aligned goals}.

For our evaluation in this paper, we used Watson Analytics Engine's Deep QA pipeline, to generate insights for some very interesting questions. The training data set belongs to two categories of process log data sets **PD** and **TD**. The questions that were asked using both these data sets are listed in Table 2.

## 3 Empirical Evaluation

Our evaluation is conducted in two phases: **Phase 1:** This is basically a preprocessing step that enables generation of effect logs, which are provided as input data to the Watson Analytics Engine (discussed in Phase 2). The `VAGAI tool` [8] annotates semantic traces from process logs with goal alignments to generate process effect logs (PD) and task effect logs (TD) respectively[2]. **Phase 2:** `Watson Analytics Engine` for generating performance and goal alignment predictions using the PD and TD data sets respectively as depicted in Table 2. For individual task level executions, the alignment predictions are at OR-refinement sub goal levels (providing alternate realization of its parent goal) for a given goal model. This is based on the accumulated effects at the completion of corresponding task execution.

The consolidated view of predictive insights as a visualization is depicted in Fig. 1. Here the performance prediction in terms of total process execution time is depicted for each observed effect at completion of a task. We started with questions of type **Q01, Q02** to generate the predictions of process performance time (in minutes) for each of the six contextual factors `DataIssues` + `AgentExplow`, `DataIssues` + `Highseverity`, `RemoteResolution` + `CustomerNew`, `RemoteResolution` + `AlertsComplete`, `SoftwareUpgrade`, `PasswordReset` + `AgentExplow`, `PasswordReset` + `Severity High` at specific semantic traces in the execution of process instances. This consolidated representation generated using the *Watson Analytics Engine* helps in predicting performance at different partial states of an instance execution. This demonstrates the impact of contexts on the execution of otherwise similar process

[2] https://www.scribd.com/document/333254045/IncidentLog.

**Table 2.** Questions to Watson Analytic Engine

| Question ID | Question text | Used data set | Question type |
|---|---|---|---|
| Q01 | Given a performance limit – what are the most commonly occurring semantic effect traces? | TD | Exploratory |
| Q02 | What are the context sets associated with processes taking high performance time? | TD | Exploratory |
| Q03 | Given the effect sequence E1E2E3, what is the probability of the process being aligned for a given goal G? | PD | Predictive |
| Q04 | Given the current effect sequence taking performance time N, what is the projected completion time of the process | PD | Predictive |
| Q05 | Given the current context, and the current effect sequence, what is the remainder of the effect sequence for a successful (goal-aligned) execution | TD | Predictive |
| Q06 | Given the current context, what will be the number of instances that are aligned with Goal G1? | PD | Predictive |
| Q07 | Given the current context, what is the probability of this instance to conclude with a specific effect sequence? | PD | Predictive |
| Q08 | Given the tickets with current effect sequence, what is the average total performance time of completion of these tickets? | TD | Predictive |
| Q09 | Given the current context, how many executed instances will be valid? | TD | Predictive |
| Q10 | Given the current effect sequence, which process designs the completed instances will be aligned with? | TD | Predictive |

execution instances. Similarly using this prediction model represented in Fig. 1, we can make predictions of performances at multiple states of process execution. This eventually can lead the organization to evaluate their resource deployment strategies, shifting to a different process design variant.

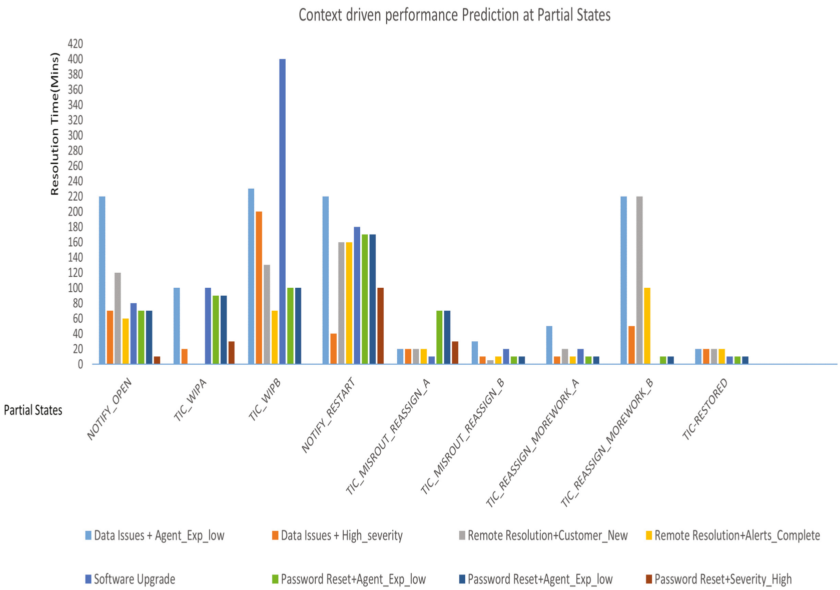


**Fig. 1.** Performance predictions at partial states

# 4 Conclusion

Organizations increasingly tend to analyze the performance drifts in day to day execution of customer and context sensitive business processes. In our proposed approach, we leverage goal correlated process variations and contextual factors mined from process log and goal correlated state transitions mined from effect logs. In our future work, we will focus on correlating dynamic run-time variations in contextual factors with shifts in goal alignment.

# References


1. Buijs, J.C.A.M., van Dongen, B.F., van der Aalst, W.M.P.: Mining configurable process models from collections of event logs. In: Daniel, F., Wang, J., Weber, B. (eds.) BPM 2013. LNCS, vol. 8094, pp. 33–48. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40176-3_5
2. Ghose, A.K., Narendra, N.C., Ponnalagu, K., Panda, A., Gohad, A.: Goal-driven business process derivation. In: Kappel, G., Maamar, Z., Motahari-Nezhad, H.R. (eds.) ICSOC 2011. LNCS, vol. 7084, pp. 467–476. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-25535-9_31
3. Heinrich, R., Merkle, P., Henss, J., Paech, B.: Integrating business process simulation and information system simulation for performance prediction. Softw. Syst. Model. **1**, 1–21 (2015)

4. de Leoni, M., van der Aalst, W.M.P., Dees, M.: A general framework for correlating business process characteristics. In: Sadiq, S., Soffer, P., Völzer, H. (eds.) BPM 2014. LNCS, vol. 8659, pp. 250–266. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10172-9_16
5. Magdaleno, A.M., de Oliveira Barros, M., Werner, C.M.L., de Araujo, R.M., Batista, C.F.A.: Collaboration optimization in software process composition. J. Syst. Softw. **103**, 452–466 (2015)
6. Maggi, F.M., Di Francescomarino, C., Dumas, M., Ghidini, C.: Predictive monitoring of business processes. CAiSE 2014. LNCS, vol. 8484, pp. 457–472. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-07881-6_31
7. Mrquez-Chamorro, A.E., Resinas, M., Ruiz-Corts, A.: Predictive monitoring of business processes: a survey. IEEE Trans. Serv. Comput. (2017)
8. Ponnalagu, K., Ghose, A., Narendra, N.C., Dam, H.K.: Goal-aligned categorization of instance variants in knowledge-intensive processes. In: Motahari-Nezhad, H.R., Recker, J., Weidlich, M. (eds.) BPM 2015. LNCS, vol. 9253, pp. 350–364. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-23063-4_24
9. Yu, Y., et al.: Case analytics workbench: platform for hybrid process model creation and evolution. In: Motahari-Nezhad, H.R., Recker, J., Weidlich, M. (eds.) BPM 2015. LNCS, vol. 9253, pp. 226–241. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-23063-4_16