Selecting Event Monitoring Points for Optimal Prediction Quality

Andreas Rogge-Solti, Nico Herzberg, and Luise Pufahl

Hasso Plattner Institute at the University of Potsdam
Prof.-Dr.-Helmert-Strasse 2-3, 14482 Potsdam
{firstname.lastname}@hpi.uni-potsdam.de

Abstract: Organizations strive to optimize their business processes in order to satisfy customer requirements and internal goals. A basic necessity in order to meet time and quality objectives is to monitor an organization's business processes. Process monitoring makes their execution more transparent and allows to react to observed deviations with corrective actions. This paper focuses on monitoring processes in manual or semi-automatic environments, where the installation of each monitoring point is costly, as it requires effort to measure and record observed progress. During process execution, the allocation of event monitoring points (EMPs) is restricted to certain positions, e.g., the termination of activities. We propose an approach for optimizing the allocation model of EMPs in order to improve the estimation quality. We implemented this approach and show its applicability in a case study of a Dutch hospital for its surgical care process.

1 Introduction

Modern companies and organizations face a challenging and ever changing market environment today. Thus, managing their business processes effectively and efficiently is essential to be competitive. A cornerstone of business process management is the monitoring of process instances. *Business process monitoring* assists in performance estimation, e.g., prediction of time until completion of a process instance or duration of certain activities [ASS11]. This technique enables the detection of performance issues, e.g., being behind schedule, so that corrective actions can be taken to finish the affected instances according to the plan and avoid deviations from planned goals and service level agreements.

Process execution ranges from completely manual, over semi-automated to fully automated enactment; the latter is a process execution according to specified models controlled by a central process engine. In automated environments, the central process engine provides execution data of the performed activities out-of-the-box usable for process monitoring. In contrast, non-automated process execution requires a separate implementation for capturing activity execution data. In several domains, the majority of process execution is still manual, such as in the healthcare domain, where the treatment processes require high flexibility and individual reactions to each patient [LR07].

In [HKRS12] an architecture is presented to make use of the sparse data that is generated while manually executing processes for monitoring purposes. In that work, the concept of event monitoring points (EMP) is introduced as well-defined places in the process model, to which process execution information can be correlated, e.g., the termination of activities. However, the information that is already available might not suffice to meet certain process monitoring requirements, e.g., prediction quality of time until completion of a process instance. In that case, additional installation of event capturing has to be considered to provide more transparency and improve prediction quality. Options for event capturing include simple stopwatches, bar-code scanners, RFID readers or other devices, but installing and running such equipment is expensive.

In this context process managers face two major questions: (1) How many EMPs have to be installed in order to reach a certain level of monitoring quality? (2) Where to optimally position the given number of EMPs to achieve best prediction quality of time until completion of a process instance? In this paper, we address the latter question and present an optimization algorithm for sequential processes. The applicability of our approach is discussed in the context of a surgical care process of a Dutch hospital. Since the operating room of a hospital is its most costly asset [MVDM95], hospitals try to maximize utilization and avoid idle times. High prediction quality for the end of the surgeries is crucial to allow rescheduling of surgeries and resources in case of deviations from the surgery schedule.

The paper is structured as follows. In Section 2 we introduce the problem and show how selection of EMPs influences the overall prediction quality of process duration. Related work is discussed in Section 3. Section 4 provides a formal description of the approach indicating where EMPs should be optimally allocated and describes the implementation of the proposed algorithm. In Section 5, we evaluate the applicability of the approach based on the use case data mentioned above. Finally, we conclude the paper and look on future work in Section 6.

2 Problem

In a manual process execution environment, a trade-off between monitoring effort and prediction quality has to be made. In order to quantify the quality of the prediction regarding the process completion, uncertainty is considered. Uncertainty is the expected deviation of the actual process end to the estimated process end, e.g., estimated process completion will be at 6 p.m. with an expected deviation of +/-1hr. In this case, the uncertainty is 1 hour. Uncertainty can be measured for example based on the observed variance in historical data or by quantifying the error of future predictions, e.g., with measuring the mean square error.

For the prediction of the process completion time, at least two monitoring points are necessary, at the beginning and at end of a process. Figure 1 illustrates the uncertainty over time when utilizing two EMPs m_1, m_2 in Fig. 1(a), resp. ten EMPs $m_1, \dots, m_{10}$ in Fig. 1(b). The distance between the EMPs is the mean duration between them. In the basic scenario with two EMPs at start and end, the monitoring effort is minimal for estimating process duration, but it gives only a very rough prediction regarding process completion time. The uncertainty in this scenario stays over the whole process execution time equal to the initial uncertainty of the mean process duration. Assuming the uncertainty is scaled, the initial maximum value is always 1 and the value at completion of a process instance is always 0.

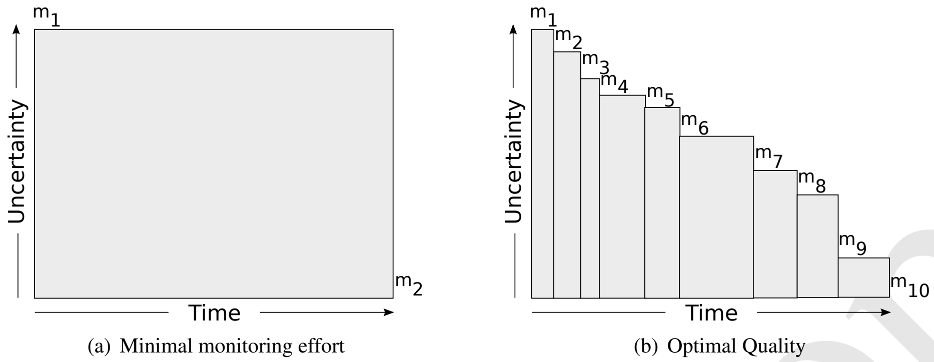


Figure 1: Distribution of uncertainty in dependence of the number of EMPs. The width of a column is the mean duration between the EMPs, the height is a measure of uncertainty, e.g., mean square error. In (a), only start and end of a process instance is measured. In (b), all possible EMPs are set up to reach the highest possible prediction quality.

Further EMPs can be installed in the process model at well-defined places. For simplicity reasons, we assume in this paper that it is possible to measure at the start of a process instance as well as at the termination of each activity. In a sequential process, the uncertainty over time regarding the process completion time is decreased by each additional EMP. Thereby, it is assumed that the activities' duration do not correlate with each other. In Figure 1(b) *all ten possible* EMPs for the process model depicted in Figure 2 are considered, i.e., EMPs at the start of a process as well as at the end of each activity. Here, the optimal prediction quality with the lowest possible uncertainty over time is achieved for this process. However, this setup will produce the highest effort of monitoring.

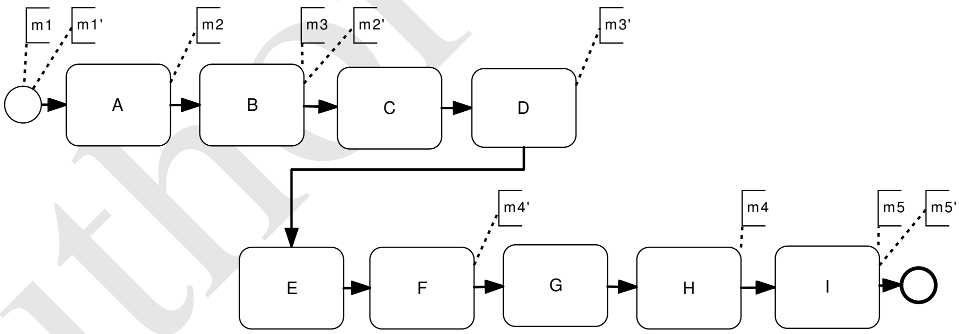
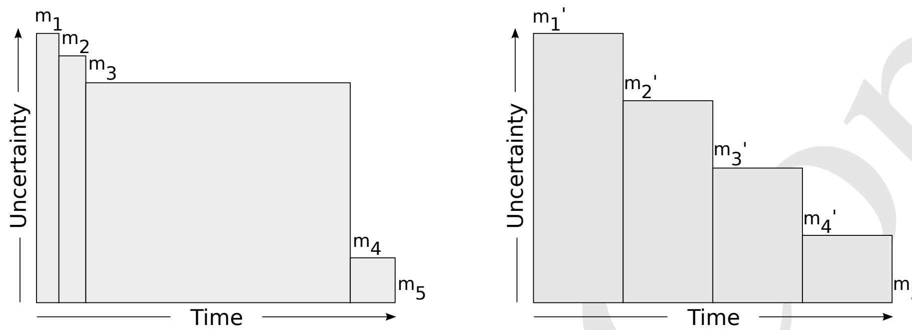


Figure 2: Process Model with differently allocated EMPs $m_1, \ldots, m_5$ (scenario 1) and $m'_1, \ldots, m'_5$ (scenario 2)

Beyond the number of EMPs, the allocation of them has also an important influence on the prediction quality. Figure 2 represents a process model where five EMPs are distributed differently. In the first scenario, the EMPs accumulate at the beginning and the end of the process ($m_1, \ldots, m_5$), whereas in the second one the EMPs are placed more homogenously over the process ($m'_1, \ldots, m'_5$). Figure 3 depicts the decreasing uncertainties with five implemented EMPs for the two mentioned scenarios in the diagrams. While comparing the overall uncertainties of the two configurations (i.e., the sum of the areas of the uncertainty bars for each EMP), it can be observed that the overall uncertainty over time is much higher in the first scenario, see Figure 3(a). There is a long period between m_3 and m_4 with high uncertainty which makes the overall uncertainty in the process higher than in Figure 3(b).



(a) Inhomogeneous monitoring (b) Homogeneous monitoring (optimal)

Figure 3: The overall uncertainty of the process, depicted in Figure 2, with five differently allocated EMPs $m_1, \ldots, m_5$ (scenario 1) resp. $m'_1, \ldots, m'_5$ (scenario 2). The four columns indicate the uncertainty of the prediction for the remaining process execution time.

Before presenting an algorithm for sequential processes which assist process managers to find an optimal allocation for a given number of EMPs to keep the overall uncertainty minimal, we present related work upon which we build.

3 Related Work

In a complementary work [RSW12], the authors described how process progress can be estimated in process models with sparse EMPs. The authors assume known probability density functions for activity durations and answer the question, to which state a process instance progressed over time on a probabilistic basis. Our approach can be used to improve the estimation quality in [RSW12] by an optimal placement of EMPs. The underlying architecture feeding the EMPs in the process models with data from scattered information throughout the IT system landscape was presented in [HKRS12]. The same architecture is assumed as the basis for the presented approach in the paper at hand.

The question, when a running process instance will be finished, is addressed by van der Aalst et al. in [ASS11] with the concept of process mining. The authors propose to use historic process execution data captured in the event logs of supporting IT systems for predicting the completion time of a running instance. Their approach starts with building up a state transition system for the respective process based on the event log. In a next step, each state of the transition system is annotated with statistical information, e.g., mean, and variance, about the remaining time until completion from this state. This information is learned from historic instances which have visited the state. The annotated transition system can then be used for making predictions for running instances. Hereby, a fully or partly automated process environment is assumed, where historic data can be simply derived from the event logs. In contrast, in this paper, we do not intend to predict completion time of running instances, we rather seek to minimize the uncertainty of the prediction under given budget constraints.

A related problem statement can be found in the research domain of project management discussing the optimal timing of control points within a project. Control points are moments in a project where control activities are conducted for measuring the actual project state against the project plan. On the one hand, control activities are important, because they allow the detection of project deviations from the planned schedule and the implementation of corrective actions. On the other hand, they produce direct costs, are time-consuming and bind resources. Hence, similar decisions have to be made: the number of control points during a project and their allocation have to be planned. Partovi and Burton [PB93] evaluated in a simulation study the effectiveness of five different control timing policies: equal intervals, front loading, end loading, random and no control. Due to the heterogeneity of projects, no clear policy could be identified to be superior. De Falco and Macchiaroli argued that individual allocation of monitoring activities is required [FM98]. Therefore, they provided a method to determine quantitatively the optimal control point distribution by defining at first the effort function of a project based on activity intensity and slack time. Then, the control points are placed around the concentration of effort.

The concept of activity intensities was also used by Raz and Erel [RE00]. They determined the optimal timing of control activities by maximizing the amount of information generated by the control points. The amount of gathered information is calculated based on the intensity of activities carried out since the last control point and the time elapsed since their execution. The authors utilized dynamic programming in order to solve their optimizing problem. It seems promising to apply that approach to the optimal allocation of EMPs in a process model as well. The difference to control points being distributed over uniform intervals, e.g, days, is that the EMPs can only be positioned at well-defined places in a process model, i.e., at the end of activities.

Another application area for optimal allocation of control points is the diagnosability of systems. A diagnosis process of a system (e.g., a microprocessor) aims at the identification of reasons for unexpected behavior [CCGDV06]. A set of sensors (observations) and a model of the system is needed to detect system components being responsible for incorrect behavior. Existing sensors divide the system into clusters, whereby the connections between components inside of a cluster are not monitored, but the connections with components outside of a cluster. Ceballos et al. [CCGDV06] present in their research work a concept to allocate a set of new sensors in order to improve the system diagnosability. Their goal is to maximize the number of monitored clusters with a given number of additional sensors. Due to the exponential complexity of this maximization problem, the authors developed a greedy algorithm. This algorithm identifies bottlenecks of the system as the best candidates for allocating new sensors. This approach was transferred to business processes in the work of Borrego et al. [BGLGC10]. Installed control points within a process can help to identify which activities are responsible for deviating behavior from the process model. For their allocation, the authors refer to the proposed algorithm in [CCGDV06]. This algorithm was already used for business processes and focuses on increasing the number of monitored activity clusters. However, a maximum number of activity clusters does not necessarily yield an optimal prediction quality. In the next section we address this issue.

4 Approach

In order to ensure high prediction quality in a certain effort frame we developed and implemented an approach including an algorithm that allocates EMPs in a process model according to the given input. The algorithm requires as input (i) a process model, (ii) data about the execution time of that model (historical records or simulated), (iii) the number of EMPs that should be allocated, and (iv) the uncertainty function that should be used for calculation. In order to describe the approach formally, we first introduce in Section 4.1 some preliminary notions. Afterwards, we describe the algorithm in Section 4.2 and present in Section 4.3 the implementation of our approach.

4.1 Preliminaries

In this paper we define the process model as a connected graph consisting of a set of activities A and control flow edges F beginning with a start event e_s and terminating with an end event e_e.

Definition 1 (Sequential Process Model) *A process model is a tuple $P = (A, F, e_s, e_e)$, where A is a set of activities, e_s is the start event and e_e is the end event of P. The flow relation is defined as $F \subseteq (\{e_s\} \cup A) \times (\{e_e\} \cup A)$ and captures the ordering constraints of the activity execution.*

In a sequential process model it holds that each node can have at most one incoming and one outgoing control flow edge, i.e.,
$(x, y) \in F \wedge (x, y') \in F \Rightarrow y = y'$ *and*
$(x, y) \in F \wedge (x', y) \in F \Rightarrow x = x'$
Hence, the flow relation yields an ordering of activities A in the process model, where $(x, y) \in F \Leftrightarrow x < y$, i.e., y can only be executed, when x has been terminated.

In a process model P event monitoring points (EMPs) can be allocated for correlating process execution information at well-defined places. In this paper, we assume that the start and end of a process are known, i.e., an EMP exists at the start of a process as well as at the end of the last activity a_n. Further EMPs can be allocated to the termination state change of all other activities of the process model.

Definition 2 (Event Monitoring Point) *Let P be a process model with the set of activities $A = \{a_1, \ldots, a_n\}$. The set of possible EMPs M is defined as the union of the EMP at the start of the process m_1, and the EMPs $m_2, \ldots, m_{(n+1)}$ capturing the termination of each individual activity $a_1, \ldots, a_n$.*

Thus, we define for a sequential process with n activities $n + 1$ possible EMPs. At these EMPs information about process execution is gained. When we observe the occurence of an event at the resp. EMP, the previously uncertain activity durations become certain at this point. Thus, the overall uncertainty is reduced from this EMP on for the remaining process duration. We want to be able to quantify this and define the mean of the remaining process duration and the uncertainty of these durations respectively.

Definition 3 (Mean of the Remaining Duration) *The remaining mean duration is a function* $mean_{\mathrm{dur}}\colon M \rightarrow \mathbb{R}_0^+$ *assigning to each EMP* $m_i \in M$ *the arithmetic mean of the durations from the time at* m_i *until the termination time of the process which is captured by the termination of the last activity in EMP* $m_{(n+1)}$*.*

When the remaining mean duration is calculated on a sample of observed values, it is usually subject to bias. This bias becomes less prominent with growing sample size due to the law of large numbers. We assume a large sample size of the historical execution data and do not consider therefore bias and statistical confidences of the observed mean. In this research work, we want to focus on the uncertainty of the predicted mean duration from the EMPs until the process end.

Definition 4 (Uncertainty of the Remaining Duration) *Let* $u_{\mathrm{dur}}\colon M \rightarrow \mathbb{R}_0^+$ *be the uncertainty function assigning a non-negative value to an EMP capturing an uncertainty measure of the remaining process duration.*

This definition does not limit to a specific uncertainty function, as there are many potential ways to measure and calculate the uncertainty of the remaining duration, e.g., by the variance (VAR). Estimation and prediction is a broad field of operational research and many measures have been introduced, e.g., Mean Square Error (MSE), Root Mean Square Error (RMSE). An overview can be found in the research work by Hyndman and Koehler [HK06].

Mean duration as well as the uncertainty is relatively scaled, so that the maximum is 1, in order to ensure comparability. With these notions, we can quantify the relative overall uncertainty in the process.

Definition 5 (Overall Uncertainty) *Let* P *be a process model with the set of activities* $A = \{a_1, \ldots, a_n\}$ *and the corresponding set of possible EMPs* $M = \{m_1, \ldots, m_{n+1}\}$*. The overall uncertainty* U *in the process is defined as:*

$$U(m_1, \ldots, m_{n+1}) = \sum_{i=1}^{n} u_{\mathrm{dur}}(m_i) \cdot (mean_{\mathrm{dur}}(m_i) - mean_{\mathrm{dur}}(m_{i+1}))$$

Note that by this definition, we interpret the overall uncertainty as the area under the stair-shaped uncertainty figures, cf. Figure 1 and Figure 3.

4.2 Algorithm for optimal placement of EMPs

In the following, the algorithm for the optimal placement of a number of EMPs is presented, such that the resulting overall uncertainty is minimal. The algorithm will closely follow the proposed approach in [RE00] for the optimal placement of control points in projects. In contrast to [RE00] in which the given control points are set at arbitrary intervals (e.g., on a per day basis), EMPs in processes can only be set at well defined positions, i.e., at the termination of an activity. Hence, the number of activities limit the maximum number of EMPs. When all EMPs are installed in a process, i.e., monitoring the start of the process and each activity's termination, the highest possible prediction quality can be achieved. The presented algorithm indicates where to implement a given number of EMPs in order to decrease the overall uncertainty of the prediction.

The problem of selecting k EMPs optimally out of n potential EMPs for overall maximal certainty of the prediction of process completion is computationally complex. There are $\binom{n}{k}$ solution candidates. The problem can be divided into computing local optimal solutions recursively. The local optimal solution only depends on the previously allocated EMP. Thus, we can store intermediate optimal solutions and skip calculating these again for the other combinations with the same previously allocated EMP. This makes a dynamic programming approach, as also proposed in [RE00], feasible and the problem can be described as follows.

The idea is to minimize the overall uncertainty U that depends on where the EMPs are installed. We do this by looking at the complementary problem of maximizing the reduction of the uncertainty. Note that the allocation of an additional EMP m_j reduces the uncertainty of the prediction by the uncertainty portion which lies between m_j and the previous EMP m_i. This decrease of uncertainty applies for the remaining mean duration $mean_{\mathrm{dur}}(m_j)$. The reduction of uncertainty $\bar{U}$ can be interpreted as the white area that complements the overall uncertainty to 1, i.e., $U + \bar{U} = 1$. Thus, the overall reduction of the uncertainty $\bar{U}$ of all allocated EMPs can be defined as:

$$\bar{U}(m_1, \ldots, m_{n+1}) = \sum_{i=2}^{n} (u_{\mathrm{dur}}(m_{i-1}) - u_{\mathrm{dur}}(m_i)) \cdot mean_{\mathrm{dur}}(m_i) \tag{1}$$

For the basic setup we always need two initial EMPs for capturing the start and the end of a process, therefore i starts with 2 in Equation (1). Thus, we formulate the problem as maximizing the reduced uncertainty $\bar{U}$ for a given number k of EMPs $(2 < k \leq n$. Let m_i denote the previous EMP. Let $\bar{U}(m_i, m_j)$ be the uncertainty removed by introducing the next EMP m_j:

$$\bar{U}(m_i, m_j) = (u_{\mathrm{dur}}(m_i) - u_{\mathrm{dur}}(m_j)) \cdot mean_{\mathrm{dur}}(m_j) \tag{2}$$

We define the maximal reduced uncertainty by allocating one additional EMP, when the last EMP is m_i, as $\bar{U}_1^*(m_i)$:

$$\bar{U}_1^*(m_i) = \operatorname*{Max}_{i<j\leq n} \{\bar{U}(m_i, m_j)\} \tag{3}$$

We are interested in finding the particular EMP that maximizes the reduction of uncertainty of the prediction when m_i is the previous EMP, i.e., we want to find the argument that is responsible for the maximum in Equation (3):

$$m_{j_1}^* | m_i = \operatorname*{arg\,max}_{i<j\leq n} (\bar{U}(m_i, m_j)) \tag{4}$$

At this stage, we can describe the optimal solution for implementing one additional EMP. However, the problem is more complex, as we are also interested in the optimal placement of two or more EMPs, given the last EMP is already set. We denote this multiplicity with an index, i.e., $\bar{U}_2^*(m_i)$ for the maximum gained certainty with two additional EMPs after the EMP m_i.

$$\bar{U}_2^*(m_i) = \underset{i<j\leq n-1}{\text{Max}}(\bar{U}(m_i, m_j) + \bar{U}_1^*(m_i)) \tag{5}$$

Further, we define the maximum reduced uncertainty $\bar{U}_k^*(i)$ for a number k of additional EMPs recursively:

$$\bar{U}_k^*(m_i) = \underset{i<j\leq n-(k-1)}{\text{Max}}(\bar{U}(m_i, m_j) + \bar{U}_{k-1}^*(m_j)) \tag{6}$$

We are further interested in the position of the next EMP in the sequence that yields the maximum decrease in uncertainty, given that the previous EMP is m_i.

$$m_{j_k}^*|m_i = \underset{i<k\leq n-(k-1)}{\arg\max}\,(\bar{U}(m_i, m_j) + \bar{U}_{k-1}^*(m_j)) \tag{7}$$

With this notation, we can formulate the problem introduced in Section 2, as to compute $\bar{U}_{(k-2)}^*(m_1)$ (cf. Equation 6), i.e., the maximum reduction of uncertainty gained by k EMPs, given that two of them measure the start and the end of the process, and return the arguments $m_{j_{(k-2)}}^*, m_{j_{(k-3)}}^*, \ldots, m_{j_1}^*$, cf. Equation (7). In order to solve this problem, the algorithm pursues the following steps:

1. Determine the set M of potential EMPs in the given process model.
2. For each $m_1, \ldots, m_{n+1} \in M$ calculate the remaining mean duration $mean_{\text{dur}}(m_i)$ until process termination based on given historical execution data.
3. Calculate uncertainty $u_{\text{dur}}(m_i)$ of remaining durations for each identified potential EMPs based on historical execution data according to the given uncertainty function.
4. Compute $\bar{U}_{(k-2)}^*(m_1)$ for the given number of requested EMPs k by using dynamic programming for searching through the $\binom{n}{k}$ solution combinations. Thereby, intermediate computed optima are stored to save time by not recomputing such solutions.

Note that the presented algorithm can be also utilized to determine the required EMPs to meet a given uncertainty threshold. Therefore, the algorithm has to be executed iteratively by incrementing k starting with 2 until the threshold is met. This is shown exemplarily Section 5.

The described algorithm is used in the implementation which is presented in the next section.

4.3 Implementation

We implemented the approach for sequential processes in ProM [DMV$^+$05]. Our developed ProM plug-in needs as input an event log of the respective process, which provides the historical execution data, consisting of the case start time and activities termination time for a set of process instances. Usually, event logs are provided in an automated process execution environment by information systems. In a manual process environment, an event log can be created by two different possibilities:

- Recording execution data over a certain period of time in the real process environment
- Performing a simulation based on an annotated process model including performance models for activities, e.g., probability density functions for durations, and collecting the simulated execution data, e.g. with CPN tools [JKW07]

No process model is needed by the implementation as it uses the ProM functionality and derives the model from the event log as proposed in [ASS11]. Furthermore, the developed plug-in has two user parameters: (1) selected uncertainty function and (2) the number of desired EMPs.

As soon as the user has picked one of the provided *uncertainty functions* $u_{\text{dur}}(M)$ (e.g., VAR, MSE, RMSE), the overall uncertainty is shown for all possible EMPs in M of the process model as the optimal stair-shaped uncertainty graph to the user, as depicted in Figure 6.

Secondly, the user can select *a number of EMPs*, which should be distributed optimally between a range from two (only start and end are monitored) to the size of possible EMPs in the process. The implementation uses dynamic programming for searching the optimal solution of the $\binom{n}{k}$ combinations as described by the algorithm in the Section 4. The resulting optimal EMPs are highlighted at the x-axis to the user and the resulting uncertainty graph based on the selection is laid behind the optimal graph in gray color for comparison. Additionally, the plug-in provides numeric values for the overall uncertainty U of the respective graphs in order to assist the user in reasoning on the required number of EMPs.

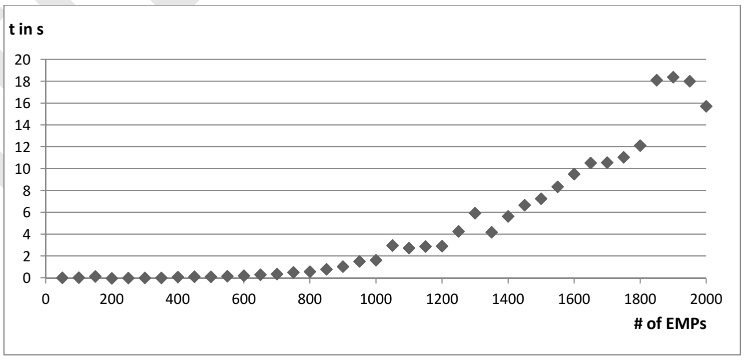


Figure 4: Algorithm runtime performance with sequential models with an increasing number of EMPs

In order to test the performance of the algorithm, we generated random stair shaped uncertainty graphs with $n = 50, 100, \ldots, 2000$ nodes and calculated the optimal position of $k = \frac{n}{10}$ EMPs on a typical laptop computer. The optimization algorithm can deal with model sizes up to 2000 nodes, but runs into memory shortages above. This is no practical issue, as process models with such dimensions should be hard to find. As shown in Fig. 4, optimal positioning in models with a size of less than 1000 possible EMPs can be computed in under two seconds.

5 Case Study

In this section, the applicability of the developed optimization algorithm is shown with a use case of the surgical care process of a Dutch hospital.

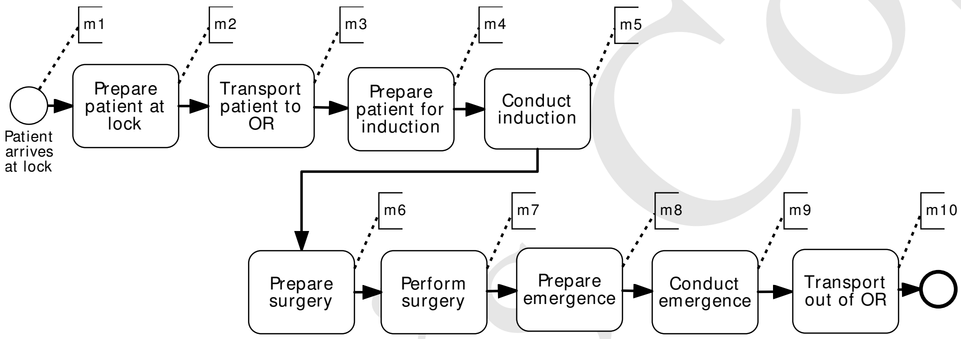


Figure 5: Surgical care process represented as BPMN diagram with annotated EMPs $m_1..m_{10}$

The process is modeled in a BPMN diagram, see Figure 5, and the currently existing EMPs, which are allocated at the start of the process and the termination of every activity, are annotated. The surgical care process starts with the arrival of the patient in the lock measured by the first EMP m_1. After the patient was prepared in the lock (m_2), he or she is transported into the operating room (m_3). The activities in the operating room start with preparing the induction of the anesthesia (m_4) followed by its conduction (m_5). As soon as all preparations for the surgery are finished (m_6), the surgical procedure is performed (m_7). Afterwards, the emergence (m_8) is prepared and the patient is emerged from his/her anesthesia (m_9). Once the patient is transported out of the operating room (m_{10}), the process ends.

All annotated EMPs are captured manually by process participants in an IT system. However, some execution data records of EMPs were empty due to human failure. Therefore, we conducted a preprocessing step in which we filtered out traces having no start or end time, because the calculation of the process mean duration and its uncertainty is not possible for such cases. In a next step, we imported the historical execution data of 935 instances for the described process into the ProM tool and applied our developed plug-in to the data. The result is shown in Figure 6. The overall uncertainty U—based on the variance here—of the prediction regarding the completion time of a surgical case for this process is 0.7583.

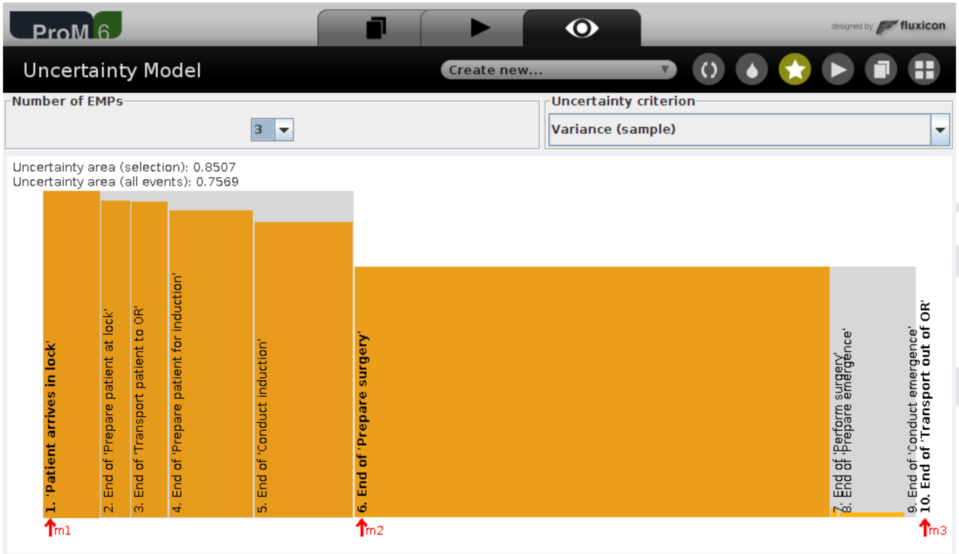


Figure 6: Screenshot of the results of the optimization algorithm implemented in ProM

The graph makes explicit that the activity with the highest uncertainty over time is the *performance of the surgery*. This can be explained by having a closer look on the execution data. The data set of the Dutch hospital contains heterogeneous types of surgeries, e.g., excision of the tonsils, breast neoplasm surgery, which show different operating times.

In a next step, the algorithm was run for the optimal allocation of three EMPs, whereby the first two have their standard position at the beginning and end of the process. The third EMP was allocated optimally by the algorithm at the end of the *surgery preparation* which is shown in Figure 6. The reduction from ten to three EMPs leads to an increase of the overall uncertainty to 0.8509. Hence, if the management of the surgery department accepts roughly 0.1 more overall uncertainty, seven monitoring points could be saved. Thus, saving efforts for manually capturing those EMPs results in slightly worse estimation quality. The participants—mainly nurses, which are in charge of recording the times—would be able to focus more on their main task and on the patients.

The following graph in Figure 7 represents the decrease of uncertainty over time by adding additional EMPs starting with two, i.e., monitoring only the start and end of a process. Note that, while every additional EMP can add to the reduction of the overall uncertainty, the first few do so with bigger impact. The highest decrease in uncertainty for this surgical care process can be gained with the third (0.15) and the fourth EMP (0.07). With every additional EMP, the decrease of uncertainty converges to the maximal value of 0.24.

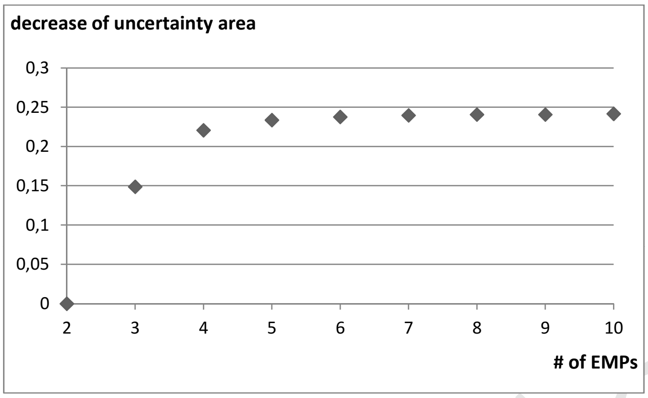


Figure 7: Decrease of uncertainty with increasing number of EMPs

This use case shows that a reduction by six to seven EMPs would lead only to a small increase of uncertainty regarding the remaining mean duration, but it would help to save efforts recording them manually. Furthermore, the algorithm supports the surgery management in deciding where a number of EMPs should be installed.

6 Conclusion and Outlook

A cornerstone of today's business process management is *business process monitoring* ensuring the quality of process execution and preventing deviations from the organization's goals and service level agreements. Especially in rather manual process enacting domains, e.g., healthcare, monitoring of processes challenges the organizations in balancing the efforts for monitoring and the quality of the predictions based on monitoring. In this paper, we presented an approach to find the optimal allocation of a given number of EMPs in sequential process models for minimizing the uncertainty of the predicted time until completion, thus gaining the best prediction quality according to a flexible uncertainty measure. The number of EMPs varies between two (i.e., start of process and end of the last activity) and all possible measurable points in the process model (i.e., start of process and termination of all activities), depending on the efforts the organization is willing to spend for process monitoring.

Our proposed optimization algorithm was implemented in the ProM tool. We showed that the runtime complexity is well manageable and works on a laptop computer for up to 2000 possible EMPs with a selection of 200 EMPs in a sequential process model that should suffice for almost any process in practice. A more important limitation of this approach is the support of sequential processes only. This is due to the fact that there exist complex relations between EMPs on parallel branches in execution that influence the uncertainty about the state of a process, as shown in [RSW12]. In this paper, we focused first on the sequential case, but we plan to lift this limitation in future work, and allow also control flow structures, e.g., exclusive and parallel gateways, or loops.

We assumed in this research work that there exists no correlation between single activities in order to assure a decreasing uncertainty over time with each additional EMP. However, positive as well as negative correlation between the duration of activities can be observed in practice. An example for a positive correlation is the surgery preparation activity which usually needs more time, if the surgery takes longer and is more complicated. A negative correlation can be observed for instance in processes with a certain deadline; activities are processed faster when previous activities took longer and deadlines get closer. Considering correlation of activities would be very valuable for predictions and is thus worth to look at in future research work.

References

[ASS11] W.M.P. van der Aalst, M.H. Schonenberg, and M. Song. Time prediction based on process mining. *Information Systems*, 36(2):450–475, 2011.

[BGLGC10] D. Borrego, M.T. Gomez-Lopez, R.M. Gasea, and R. Ceballos. Determination of an optimal test points allocation for business process analysis. In *Network Operations and Management Symposium Workshops (NOMS Wksps), 2010 IEEE/IFIP*, pages 159–160. IEEE, 2010.

[CCGDV06] R. Ceballos, V. Cejudo, R. Gasca, and C. Del Valle. A topological-based method for allocating sensors by using CSP techniques. *Current Topics in Artificial Intelligence*, pages 62–68, 2006.

[DMV+05] B.F. van Dongen, A.K.A. de Medeiros, H.M.W. Verbeek, A.J.M.M. Weijters, and W.M.P. van der Aalst. The ProM framework: A new era in process mining tool support. *Applications and Theory of Petri Nets 2005*, pages 1105–1116, 2005.

[FM98] M. de Falco and R. Macchiaroli. Timing of control activities in project planning. *International Journal of Project Management*, 16(1):51–58, 1998.

[HK06] R.J. Hyndman and A.B. Koehler. Another look at measures of forecast accuracy. *International Journal of Forecasting*, 22(4):679–688, 2006.

[HKRS12] N. Herzberg, M. Kunze, and A. Rogge-Solti. Towards Process Evaluation in Non-automated Process Execution Environments. In *Proceedings of the 4th Central-European Workshop on Services and their Composition, ZEUS*, 2012.

[JKW07] K. Jensen, L.M. Kristensen, and L. Wells. Coloured Petri Nets and CPN Tools for modelling and validation of concurrent systems. *International Journal on Software Tools for Technology Transfer (STTT)*, 9(3):213–254, 2007.

[LR07] R. Lenz and M. Reichert. IT Support for Healthcare Processes – Premises, Challenges, Perspectives. *Data & Knowledge Engineering*, 61(1):39–58, April 2007.

[MVDM95] A. Macario, T.S. Vitez, B. Dunn, and T. McDonald. Where are the costs in perioperative care?: Analysis of hospital costs and charges for inpatient surgical care. *Anesthesiology*, 83(6):1138, 1995.

[PB93] F.Y. Partovi and J. Burton. Timing of monitoring and control of CPM projects. *Engineering Management, IEEE Transactions on*, 40(1):68–75, 1993.

[RE00] T. Raz and E. Erel. Optimal timing of project control points. *European Journal of Operational Research*, 127(2):252–261, 2000.

[RSW12] A. Rogge-Solti and M. Weske. Enabling Probabilistic Process Monitoring in Non-automated Environments. In *Enterprise, Business-Process and Information Systems Modeling*, volume 113 of *LNBIP*, pages 226–240. Springer, 2012.