International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 08 Issue: 01 | January - 2024 SJIF Rating: 8.176 ISSN: 2582-3930

Automated Software Process Discovery and Real-Time Monitoring

Vinay Somarapalli
CGI Information Systems and Management Consultants Private Limited
Bengaluru, Karnataka, India
vinay.s@cgi.com

Nageswara Vesepogu
CGI Information Systems and Management Consultants Private Limited
Bengaluru, Karnataka, India
nageswararao.vesepogu@cgi.com

ABSTRACT

Process monitoring is the concept based on managing the state of a business processes executed in an enterprise software. The named status and execution time are two important parameters to infer the state of a process in real-time. An execution of the process end to end is called as process instance. This paper covers monitoring process instances in real-time based on the data metric generated from event logs, where data metrics of a Business Process Models (BPM) – a Weighted Directed Acyclic Graph (WDAG) represents process model with its heuristics such as complete execution time of process, execution time between steps/activities and the status. Firstly, the process model is discovered either by manually or automatically from the events that are recorded from the logs using unsupervised learning techniques and heuristics of different steps of a process are modeled using statistical methods.

Both process model and its heuristics are deployed to help process engineers to understand the path of executions of a business process, its state and performance in real-time. Path of executions involves most successful path, longest and shortest paths, error state if the process instances abruptly ends in between, state of failure etc., On the performance front the range of time taken to complete the process, for each step, most and least time consuming, identifying the steps taking more time than usual etc., Process engineers/users are notified if abnormalities are observed.

KEY WORDS

Process Mining, Process Monitoring, BPM, PM4PY, Standard Deviation (SD)

DISCLAIMER: The research, views and opinions expressed in this paper are solely of the authors and do not represent the views of or is in any way endorsed by the author's employer (CGI Information Systems and Management Consultants Private Limited or any of its affiliates). The author's employer shall not be liable for any damages or losses whatsoever, arising out of or in connection with this paper.

1 Introduction

In an enterprise software, the business transactions are defined with a set of processes or sequence of activities which normally follows a start to end. During the course of execution, every event of a process is recorded in log files and sourced to mining database to maintain detailed trails called event logs [1]. These event logs consist of traces and each trace with sequence of activities performed, execution time stamp, resource handling, time taken, cost incurred and other informative parameters. These recorded event logs are then extracted from the database and converted into Extended Event Streams XES [2], a standardized format to explore the hidden insights using the process mining techniques i.e. discovery, conformance and extension [3]. With the help of explored insights, the end user or business analyst would be able to identify the bottlenecks causing the process to languid, deviations from the sequence of actions and other useful information and help to come up with appropriate measures like areas required to be improved, ramp up or down the resources and mitigate the financial impact.

Automated Process Mining/Discovery and real-time monitoring are most valuable steps to the business to improve the sequence of steps that adds value to the end customers. It helps the engineering teams to document, uncover best practices, notify if any deviations, an opportunity for improvements [4].

The process monitoring is distinguished into, a) active monitoring which is concerned with "real-time" propagation of relevant data concerning the enactment of business processes, such as the status or the execution time; and b) passive monitoring which delivers information about process instances upon request [4]. In either case, there should be some reference data, often called as data metric or simply metric to compare with the newly generated process instance for status and execution time.

In this paper, we briefly discuss about the approach to identify critical business processes and more about real-time monitoring. And, we define the term metric is considered to be a graph data of efficient BPM[1] and corresponding statistical data as heuristics, to examine the status and execution time of a process instance respectively.

From figure 1, the metrics are generated from the historical event logs using process mining techniques and statistics. Metrics are used to build statistical models in the form of weighted directed acyclic graph. The directed acyclic graph represents the process model and weights between nodes represent the execution time.

[1] The process model defines the best sequence of activities is or to be followed to make the business process compliant and free from bottle necks, economical losses or from other forms of problems. i.e. generating the directed acyclic and weighted graph.

The generation of metric plays a vital role in process monitoring. The efficient process model is defined in 2 ways. First, manually by the analyst with good business knowledge dealing with, or, Second, through automated process discovery [5]. In the latter case, the F1 score or F-score defined in [3, 5], which is the harmonic mean of the two measurements i.e. fitness and precision as a single metric to determine the efficient process model, $F1\ score = 2 * \frac{(fitness * precision)}{(fitness + precision)}$. Then, the statistical analysis performed on event logs to generate, 1) average throughput time and its standard deviation, 2) average and its standard deviation of time taken to complete the specific activity, which will be discussed in the coming sections.

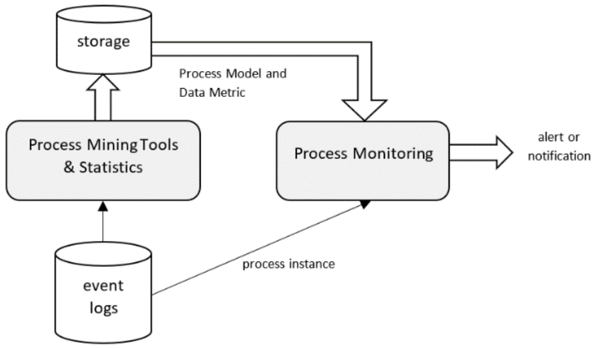

Figure 1: Overview of Process Monitoring Implementation Approach

The generated metric is then stored into the storage device in a convenient format and used to compare during the process monitoring against the process instance received at real-time a) for any deviation in the activity, and b) for the breach in time beyond 3 * Standard Deviation (3SD). Thereafter, the alert or notification is generated to end users for quick action on the process instance, on which abnormalities observed.

PM4PY framework [6] and supporting libraries are used predominantly for modeling.

The rest of the paper is organized is as follows. Section-2 briefs on dataset we worked on, Section-3 specifies approach followed to generate metrics, Section-4 details on the evaluation performed with the metric, monitoring result in Section-5 and finally we conclude in Section-6.

2 Setup and dataset

The whole analysis and generation of metric required for monitoring is developed using python and PM4PY framework [6] for modelling and mining and several other compatible libraries.

As part of model building, the event logs of business processes of an enterprise applications are extracted and converted into Extended Event Stream (XES) format, which is the standard format to source events to PM4PY Library for the cross verification. The dataset published in [7] is used to explain the model parameters and verification.

3 Metric Generation

In active process monitoring, it is very important to know the status and execution time of a process instance, this would help to check for deviation and breach in time. By doing so, business will get ample time to act and adopt quick measures and thus save time or from any economic losses.

To perform process monitoring in real-time, we need a metric to compare with new process instance(s) generated by the system. To generate a data metric, we take a simple and efficient approach - 1) process discovery to come up with efficient process models, and 2) statistical methods to get accepted execution time for an activity or whole process.

Prior to the generation of process model, the event logs are preprocessed to prune loops and concurrent activities [8, 9].

3.1 Discovery of Business Process Model (BPM)

Discovering the process from the event logs, we might end up with spaghetti-like model. But for originations to be very beneficial the lasagna-like model is preferred and said to be efficient BPM. The efficient process model is discovered using Inductive miner algorithm from PM4PY either by manually or automatically,

3.1.1 Manual: Usually, the analyst with a good business knowledge exploits the event logs and comes up with several factors like different ways the process is performed (called variants), determining their frequency, bottlenecks causing the process to slowdown leading to delay in completion and so on. Based on the derived factors, the analyst will create the BPM either a) through varying the noise threshold [6] or, b) by selecting top performing variants and merging them to define a standard process [6, 9], and sometimes, the analysts are allowed to edit the final version of graph to introduce, eliminate or bypass an ad hoc activity(ies).

3.1.2 Automatic: Automated discovery process operations are applied on the event logs sourced from an enterprise application and resulting the business process model as output [5]. During the operation the F1-Score or F-Score [3, 5], used to determine the BPM based on the best score obtained,

$$F1\ score = 2 * \frac{(fitness * precision)}{(fitness + precision)}$$

To begin with, the following approach is followed to capture the efficient BPM as given below,

1. Initialize the noise threshold,
2. Generate the process model based on the noise threshold from any of the discovery technique,
3. Perform conformance checking using token replay on the generated model,
4. Evaluate fitness, precision and then calculate F1 score,
5. Vary the noise threshold (varied from 0 to 1 with step size of 0.5) and repeat step 2 and 3, and
6. Finally, the BPM selected with the best F1 score.
From either of the approach, the directed acyclic graph is generated for monitoring the status of instances.

3.2 Statistical Data

After generation of BPM from the above steps, the corresponding execution time – the overall time taken to complete the process and the time took to complete each activity is evaluated.

In general, the average is considered and estimated for future process instances. But, there maybe be a cases or instances in the past, which might have taken more than the defined time aka. anomalies, these instances will affect the calculation of average and SD leading to extra execution time. So, these exceptional cases are handled before creating the statistical data for monitoring reference.

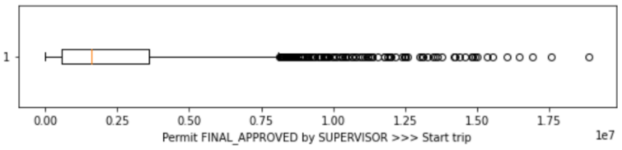

Figure 2: Box Plot for Execution Time (seconds) from “Permit FINAL_APPROVED by SUPERVISOR” and “Start trip” activity

From figure 2, when the execution time between 2 activities plotted on to the box plot, it is observed that certain processes with exceptional or delayed falling beyond the max whisker which are marked and anomalies. These anomalies are treated as outliers and ignored from calculation. We select the time falling between max and min percentile, say 80 and 20 respectively [10]. The percentile is decide depending on the severity of business for which the metric is generated.

The mean (Average) and standard deviation (SD) are calculated from the values falling between max and min percentiles and these values acts as weights for BPM generated earlier,

$$mean_{time} = \frac{1}{n}\sum time_i \ in \ seconds)$$

$$sd_{time} = \sqrt{\frac{\sum(time_i - mean_{time})^2}{n}}$$

4 Monitoring

In the previous section, we have discussed about the generation of data metric. And now, we will discuss on the approach applying metric to monitor the process instances generated in real-time.

From figure 1, the event logs are stored and sourced from event database. Start and End event and its detailed trial are sourced to modeling module to build process models. The process models and along with its heuristics is deployed to monitor the running instance of a process in real-time. The process monitoring module will register and alert and trigger a notification to service engineers if any abnormalities is observed against the data metric generated. This would help the business to revert with action to quickly adopt special measures that can mitigate from eventual consequences.

4.1 Status

It is good know the status of every instance to check for deviation is observed and correct it immediately or analyze the reason. During the evaluation process, the following steps are performed based on graph traversal with breadth first concept,

1. Read the process instance data and data metric,
2. Start with first activity,
3. Get the list of next activity(ies) of the current activity from the BPM defined in data metric,
4. Get the next activity from the process instance,
5. Compare the activities extracted from step 3 and 4,
6. Compare, If the next activity from step 4 available in the activities in step 3?
   i. Yes,
      - go to step 3
   ii. No,
      - notify the end or business user

4.2 Execution time

Similar to monitoring for deviation in status, every activity will have to be completed in the defined, so it also important to monitor overall execution time and time of previous activities took to attain completion status.

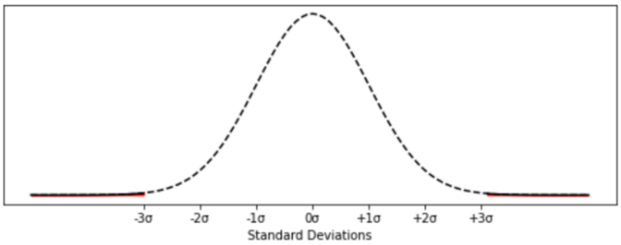

Figure 3: Plot of 3 Standard Deviation

In our paper, we define ±3σ (3SD) as threshold for testing. The realistic threshold is defined based on the type of business. Suppose, if the financial institutions need a strict compliance the threshold can be reduced to ±2σ or ±1σ. If the activity execution time falls beyond the threshold then the end users or business are notified.

5 Results

To simulate the process monitoring, we have created the synthetic data from [7] as reference and processed it as real-time instance against the data metric generated from the steps mentioned in section 3. while monitoring in real-time, we observe the anomalies in sequence of execution of activities and breach in time, base on we describe following parameters,

- **Efficient BPM**: the process models falls in-line and aligned by the business which mitigates the risks,
- **Compliant Process Instance**: the instance is free from deviation of sequence activities and acted/performed within the ±3σ window and complied to the process models, and
- **Non-Compliant Process**: deviation in the sequence of activity performed, that is, the anomaly detected or the time to taken to perform the activity is beyond the ±3σ. These type of process instances might require immediate attention by the business.

See appendix for more details.

6 Conclusion

The statistical approach for process monitoring is simple and efficient, helps the business and organizations to define and monitor the process on real-time basis. Defining BPM is crucial, but simplified with automated BPM discovery approach. The severity level of monitoring i.e. from low to critical for the business can be decided based on the percentile during data metric generation and level of standard deviation during monitoring the process instances. The notification from the abnormalities, will help them to revert into action immediately and implement quick measures to mitigate the eventual consequences. The proposed approach is easy to deploy and fast in monitoring even when many process instances generated by the application.

REFERENCES

[1] W. van der Aalst, T. Weijters and L. Maruster, *Workflow mining: discovering process models from event logs*, in IEEE Transactions on Knowledge and Data Engineering, vol. 16, no. 9, pp. 1128-1142, Sept. 2004.

[2] W.M.P. van der Aalst, B.F. van Dongen, J. Herbst, L. Maruster, G. Schimm, and A.J.M.M. Weijters. *Workflow Mining: A Survey of Issues and Approaches. Data and Knowledge Engineering*, 47(2):237–267, 2003.

[3] W. van der Aalst, *Process Mining - Data Science in Action*. Springer, 2016

[4] De Medeiros A.K.A. et al. (2007) *An Outlook on Semantic Business Process Mining and Monitoring*. In: Meersman R., Tari Z., Herrero P. (eds) On the Move to Meaningful Internet Systems 2007: OTM 2007 Workshops. OTM 2007. Lecture Notes in Computer Science, vol 4806. Springer, Berlin, Heidelberg.

[5] Anatoliy Batyuk and Volodymyr Voityshyn *Software architecture design of the information technology for real-time business process monitoring*, Econtechmod. An International Quarterly Journal – 2018, vol. 07, no. 3, 13 – 22

[6] A. Augusto et al., *Automated Discovery of Process Models from Event Logs: Review and Benchmark* in IEEE Transactions on Knowledge and Data Engineering, vol. 31, no. 4, pp. 686-705, 1 April 2019.

[7] *PM4PY*, https://pm4py.fit.fraunhofer.de/

[8] *BPI Challenge 2020*, https://data.4tu.nl/collections/_/5065541/1

[9] Martin N., Martinez-Millana A., Valdivieso B., Fernández-Llatas C. (2019) *Interactive Data Cleaning for Process Mining: A Case Study of an Outpatient Clinic's Appointment System*. In: Di Francescomarino C., Dijkman R., Zdun U. (eds) Business Process Management Workshops. BPM 2019. Lecture Notes in Business Information Processing, vol 362. Springer, Cham.

[10] A. Augusto, R. Conforti, M. Dumas and M. La Rosa, *Split Miner: Discovering Accurate and Simple Business Process Models from Event Logs*, 2017 IEEE International Conference on Data Mining (ICDM), 2017

[11] Devore, Jay L. 2016. *Probability and statistics for engineering and the sciences*.

[12] Dumas, M., La Rosa, M., Mendling, J., & Reijers, H. A. (2018). *Fundamentals of business process management (2nd ed.)*. Springer Berlin.

Appendix

The results of the proposed real-time monitoring process are visualized below,

For the generated metrics, we define,

- **Efficient BPM**: the BPM approved by business are the sequence of activities in green and dashed lines
- **Compliant Process Instance**: the process instance followed the sequence of execution steps and depicted in green.
- **Non-Compliant Process Instance:** the anomaly detected i.e. new activity is performed and no knowledge on succeeding activities or performed activity instead of performing either of specific activity. Or, more time taken to complete the previous activity. And these non-compliant actions are highlighted in red.

In the figure A, we have identified the anomaly called "Declaration being REVIEWED by SUPERVISOR" after "Declaration APPROVED by BUDGET OWNER", this would help the business to act appropriately and swiftly.

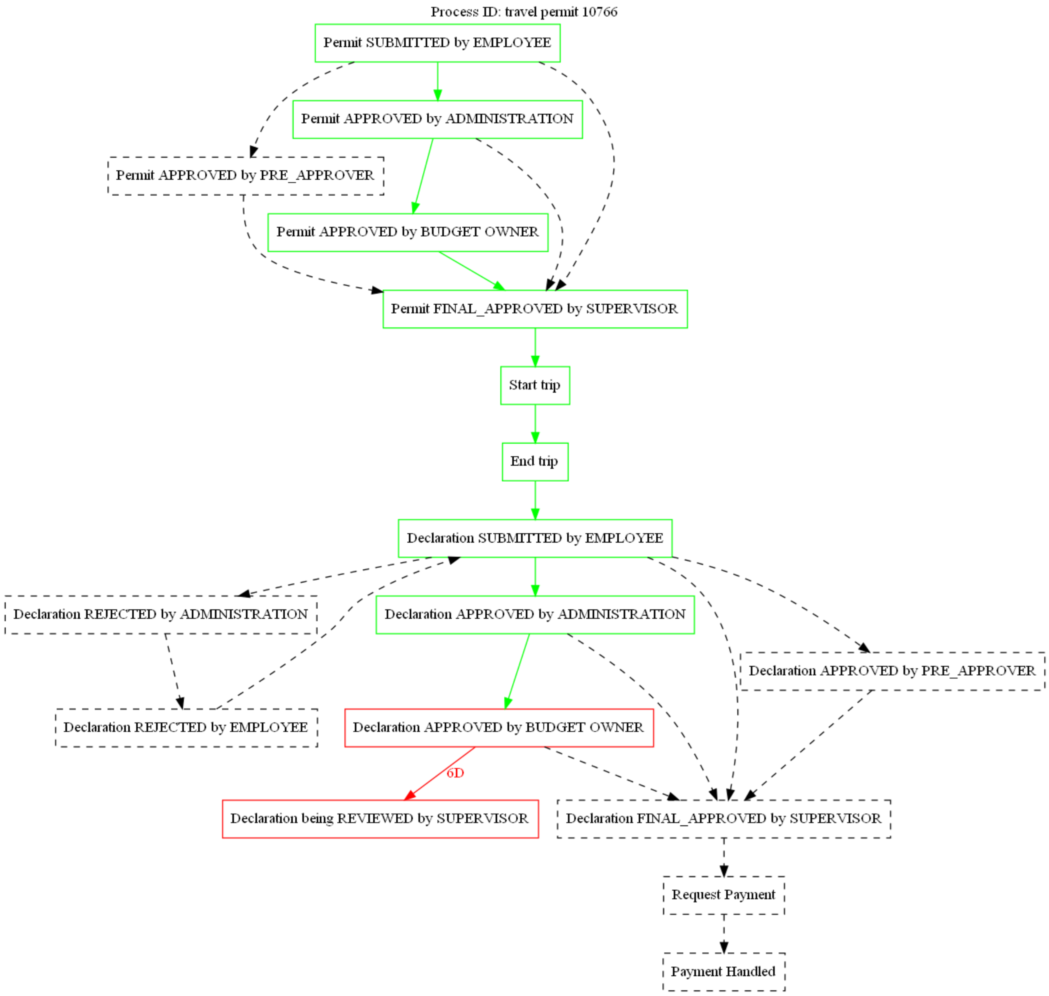
Process ID: travel permit 10766
Permit SUBMITTED by EMPLOYEE
Permit APPROVED by ADMINISTRATION
Permit APPROVED by PRE_APPROVER
Permit APPROVED by BUDGET OWNER
Permit FINAL_APPROVED by SUPERVISOR
Start trip
End trip
Declaration SUBMITTED by EMPLOYEE
Declaration REJECTED by ADMINISTRATION
Declaration APPROVED by ADMINISTRATION
Declaration APPROVED by PRE_APPROVER
Declaration REJECTED by EMPLOYEE
Declaration APPROVED by BUDGET OWNER
6D
Declaration being REVIEWED by SUPERVISOR
Declaration FINAL_APPROVED by SUPERVISOR
Request Payment
Payment Handled

Figure A: Deviation Observed the Sequence of Activity Performed

Similarly, in figure b, the time breach had been observed in 3 different places i.e. the taken complete the current activity and reach to next activity. This observation helps the business to optimize the resources as required.

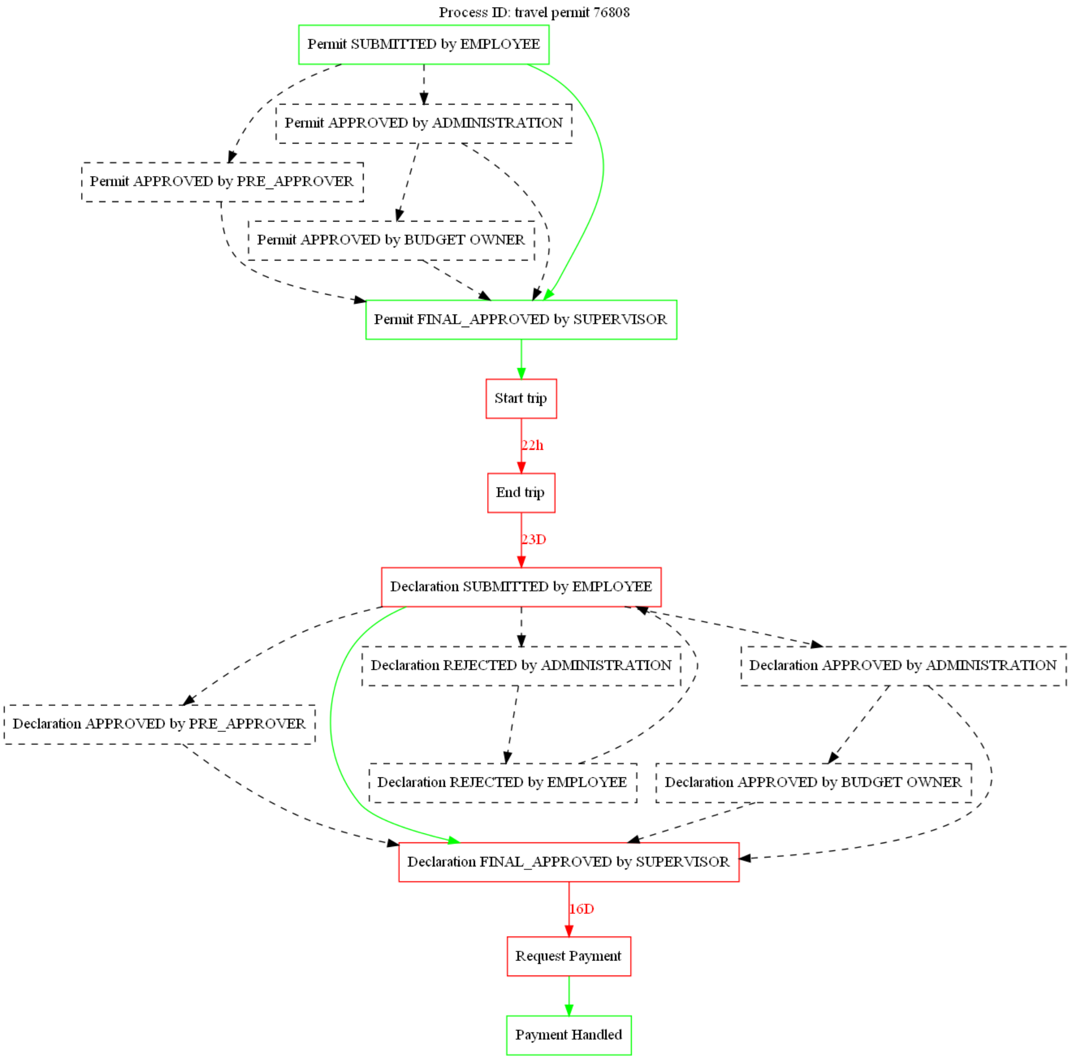
Process ID: travel permit 76808
Permit SUBMITTED by EMPLOYEE
Permit APPROVED by ADMINISTRATION
Permit APPROVED by PRE_APPROVER
Permit APPROVED by BUDGET OWNER
Permit FINAL_APPROVED by SUPERVISOR
Start trip
22h
End trip
23D
Declaration SUBMITTED by EMPLOYEE
Declaration REJECTED by ADMINISTRATION
Declaration APPROVED by ADMINISTRATION
Declaration APPROVED by PRE_APPROVER
Declaration REJECTED by EMPLOYEE
Declaration APPROVED by BUDGET OWNER
Declaration FINAL_APPROVED by SUPERVISOR
16D
Request Payment
Payment Handled

Figure B: Observed Deviation and Defiance in Time