From the Proceedings of the NSF Workshop on Workflow and Process Automation, Athens, Georgia, May, 1996

Discovery and Validation of Processes

Jonathan E. Cook and Alexander L. Wolf
Department of Computer Science
University of Colorado
Boulder, CO 80309 USA
{jcook,alw}@cs.colorado.edu

Abstract

We are developing automated data analysis methods to help organizations understand, improve, and control their processes. The methods use process event data to reveal significant behavioral characteristics of processes. The two main areas we are addressing are process discovery and process validation. Process discovery uses event data to create formal models of processes that can be used to understand an organization's processes. Process validation uncovers discrepancies between an organization's intended processes and the processes actually followed. We have implemented prototype discovery and validation tools within a common event-data analysis framework called Balboa.

Introduction

Managers today are making critical technical and business decisions based on their understanding of the processes instituted in their organizations and on their belief that those processes are indeed being carried out. But these foundations for decision making can be unreliable if they are not drawn from hard data describing the processes.

Fortunately, a substantial body of work is growing in the area of process data collection and analysis (e.g., [1, 4, 5]). In particular, both manual and automated methods are being developed to collect data describing the significant events occurring during the execution of processes by an organization. In concert, methods are being developed to analyze those data to uncover the key characteristics and anomalous behaviors of processes.

Our work contributes to this effort in two major ways. First, we are developing methods for automated *process discovery*, which is the use of event data to generate formal models of processes [3]. The goal is to create models with which one can reliably reason about the suitability of changes. Second, we are developing methods for automated *process validation*, which is the use of event data to uncover discrepancies between an organization's intended processes and the processes actually followed [2]. Deviations from intended processes occur for many reasons, both good and bad. But in all cases, deviations indicate places where something important may be happening. The process discovery and validation methods have been implemented as tools within a more general event-data analysis framework called Balboa.

Process Discovery

The process discovery methods analyze streams of event data collected from executing processes to infer nondeterministic state machine models of behavioral patterns. These patterns represent the major sequencing and iteration of activities within the process, as well as the major decision or branching points within the process.

To date we have implemented three methods for process discovery. They span a range from algorithmic to statistical.

KTail is a purely algorithmic method that places each position in an event stream into a class, determined by the k-length event sequences following that position. Positions having the same following sequences are considered equivalent and so are designated as the same state in the inferred process model.

RNet is a purely statistical, neural-net method. The net is trained to predict the next event for each position in the training event stream. Once trained, its internal activation behavior is examined to determine the state machine that has been learned.

Markov is a combination of algorithmic and statistical techniques. It builds tables of probabilities of event subsequences, using them to construct a state machine model.

The methods have been implemented as a tool within the Balboa analysis framework. Each method incorporates tuning parameters to allow a certain degree of guidance by process engineers. Thus, the discovery tool complements a process engineer's knowledge, providing empirical analysis in support of experience and intuition.

Figure 1 shows a screen shot from our discovery tool. A form-based window allows one to choose a discovery method, set its tuning parameters, and choose a data set to analyze. The tool then runs the discovery method, which produces a textual representation of a state machine. This textual representation is converted into a graphical visualization and displayed in another window. The process model display window provides editing functions that allow a process engineer to refine the process model, by splitting and merging states in the model.

Process Validation

Process validation measures the correspondence between a formal model of intended process behavior and the actual behavior exhibited by a process. The methods we use are based on string difference metrics, which count the number of insertions, deletions, and substitutions of characters needed to transform one string into the other.

For process validation, the characters in a string represent individual process events. The two strings being compared are, respectively, the stream of events predicted by the model and the stream of events collected from the process. Since the model can be nondeterministic, we use model-searching techniques to produce the event stream that is "closest" to the captured event stream. Viewing the model event stream as the target string and the collected event stream as the string to be transformed, insertions are interpreted as activities that were missed in the actual process, while deletions are interpreted as extra activities performed in the actual process that were not called for by the model.

To date we have implemented three methods for process validation.

REC is the Recognition metric. It is a straightforward metric that has just two values, true and false. The value is true if the event streams exactly match and is false otherwise.

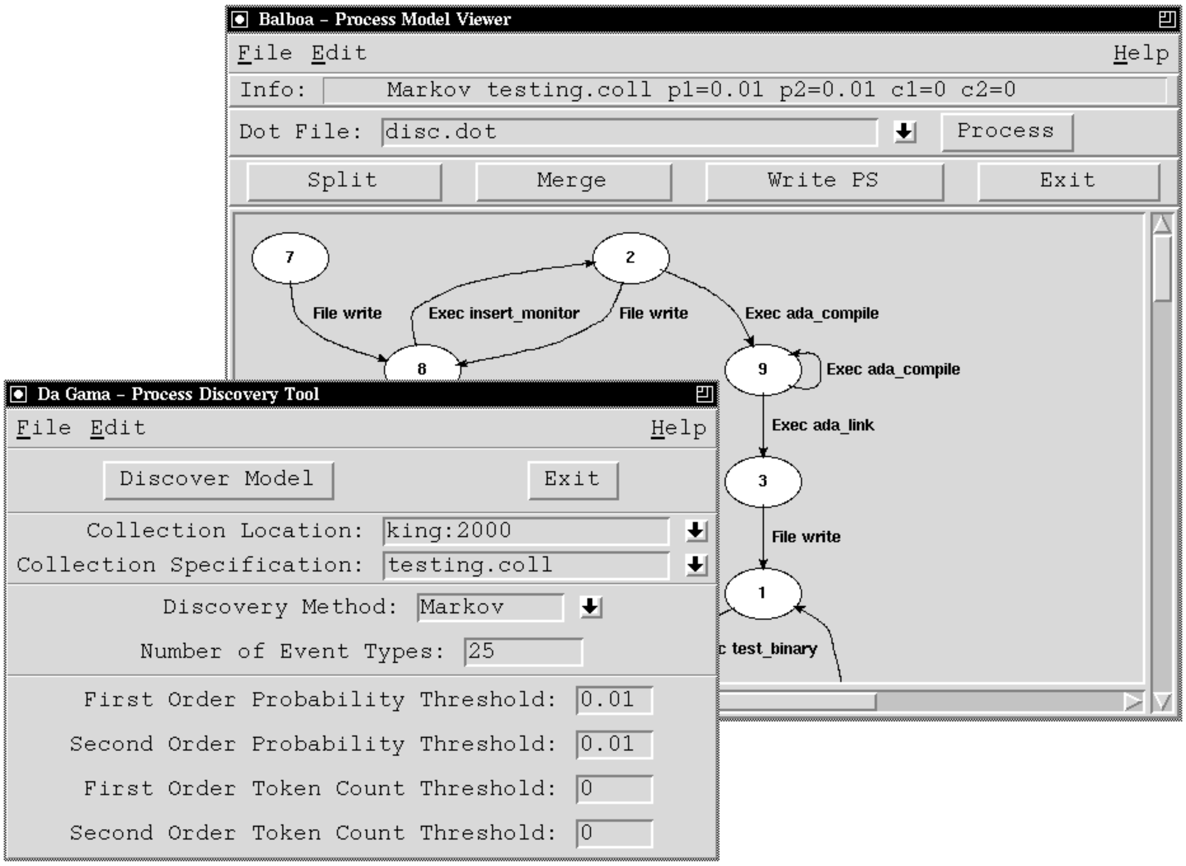

Figure 1: Screen Shot from the Discovery Tool.

SSD is the Simple String Distance metric. It calculates a simple weighted count of the number of insertions and deletions, and then normalizes this value to the length of the execution event stream.

NSD is the Non-linear String Distance metric. It extends SSD by taking into account blocks of insertions and deletions. A block represents a single, longer deviation, so rather than just counting single-event transformations, as in SSD, NSD calculates its correspondence measure by applying a parameterized exponential function to the block lengths.

The methods have been implemented as a tool within the Balboa analysis framework and, like the discovery tool, incorporates tuning parameters to allow a certain degree of guidance by process engineers.

Figure 2 shows a screen shot from our validation tool. The tool has calculated a measure of deviation for a model and data-set pair. It displays the two streams side-by-side, allowing interactive browsing of their highlighted differences.

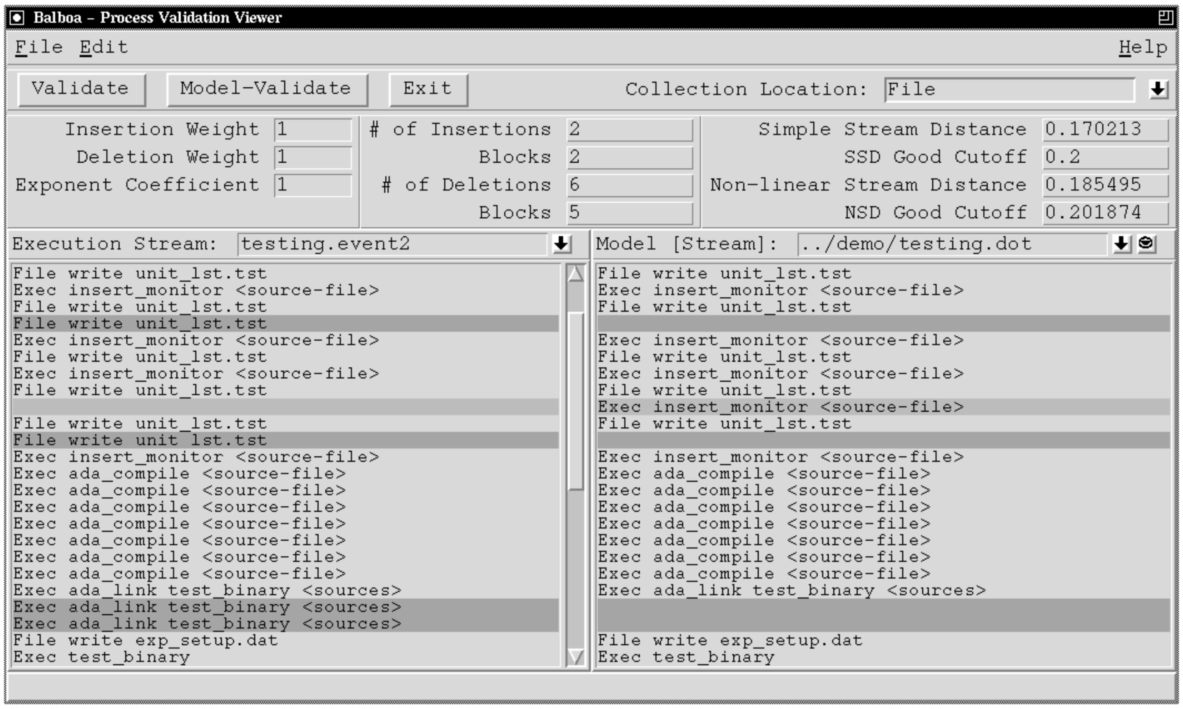

Figure 2: Screen Shot from the Validation Tool.

Conclusion

For an organization's processes to be improved, they must first be understood. The process discovery and validation methods that we are developing help raise confidence in the soundness of this understanding by providing timely analysis of concrete data collected from the processes.

Further work needs to be done in the area of process data collection. In particular, we must find less costly means of collecting process data on a continual basis. Current methods either use human-intensive methods, such as direct observations, or use automated methods, such as on-line monitoring. The former are high cost and fallible, while the latter bias the data toward on-line activities [6]. Both are subject to questions of privacy.

Likewise, further work needs to be done in the area of process data analysis. We are currently pursuing this in two directions. First, we are developing additional tools within the Balboa framework that will provide cyclicity analyses—that is, they will look for overly long or unnecessary cycles of activity. Second, we are beginning to develop methods for correlating process problems with product problems. In particular, we are nearing completion of a study conducted in collaboration with Larry Votta of AT&T Bell Laboratories in which process deviations are being correlated with product change requests.

Acknowledgments

This work was supported in part by the National Science Foundation under grant CCR-93-02739 and by the Air Force Material Command, Rome Laboratory, and the Advanced Research Projects Agency under Contract Number F30602-94-C-0253. The content of the information does not necessarily reflect the position or the policy of the U.S. Government and no official endorsement should be inferred.

References

[1] M.G. Bradac, D.E. Perry, and L.G. Votta. Prototyping a Process Monitoring Experiment. *IEEE Transactions on Software Engineering*, 20(10):774–784, October 1994.

[2] J.E. Cook and A.L. Wolf. Toward Metrics for Process Validation. In *Proceedings of the Third International Conference on the Software Process*, pages 33–44. IEEE Computer Society, October 1994.

[3] J.E. Cook and A.L. Wolf. Automating Process Discovery through Event-Data Analysis. In *Proceedings of the 17th International Conference on Software Engineering*, pages 73–82. Association for Computer Machinery, April 1995.

[4] S.L. Pfleeger and H.D. Rombach. Special Section on Measurement-based Process Improvement. *IEEE Software*, 11(4):9–85, July 1994.

[5] A.L. Wolf and D.S. Rosenblum. A Study in Software Process Data Capture and Analysis. In *Proceedings of the Second International Conference on the Software Process*, pages 115–124. IEEE Computer Society, February 1993.

[6] A.L. Wolf and D.S. Rosenblum. Process-centered Environments (Only) Support Environment-centered Processes. In *Proceedings of the 8th International Software Process Workshop*, pages 148–149, March 1993.