Chapter 61
An Integrated Infrastructure Using Process Mining Techniques for Software Process Verification

Tuğba Gürgen
Computer Engineering Department of Hacettepe University, Turkey

Ayça Tarhan
Computer Engineering Department of Hacettepe University, Turkey

N. Alpay Karagöz
Software Management Program of Middle East Technical University, Turkey

ABSTRACT

The verification of process implementations according to specifications is a critical step of process management. This verification must be practiced according to objective criteria and evidence. This study explains an integrated infrastructure that utilizes process mining for software process verification and case studies carried out by using this infrastructure. Specific software providing the utilization of process mining algorithms for software process verification is developed as a plugin to an open-source EPF Composer tool that supports the management of software and system engineering processes. With three case studies, bug management, task management, and defect management processes are verified against defined and established process models (modeled by using EPF Composer) by using this plugin over real process data. Among these, the results of the case study performed in a large, leading IT solutions company in Turkey are remarkable in demonstrating the opportunities for process improvement.

INTRODUCTION

Processes that are used in developing a software product have a strong effect on product quality (Humprey, 1995). Software companies that adopt the process-oriented management practices are expected to establish process definitions and objectives, apply the processes as specified, verify the process

implementations according to process specifications, and improve the processes in accordance with the verification results. If the software process cannot be verified, it means that source of process data cannot be detected clearly and therefore, process data cannot be used for quantitative management of the software process, which results in vague interpretations (Tarhan & Demirörs, 2012).

Quantitative management of the process is based on the principals of statistical process control (Wheeler, 1995). A process that can be controlled is a stable process and the results of this process can be predicted. A stable process must be permanently executable. The desirable before quantitative analysis of the process is, to assure that process data comes from a uniform source, in other words, the process is implemented consistently. However, process data can be originated from many sources and each of them must be considered separately. If a process is not stable, the consistency of the process must be examined in order to derive correct results (William & Robert, 1997). Consistency control aims to explore the root causes of the instability of the process. Whether the process is verified against process specifications is important for identification of different sources that generate process data, and elimination of impacts that lead to process variations. A process is candidate to quantitative management only after its implementations are consistent in itself.

Software process reference models (CMMI Product Team, 2010), (ISO/IEC, 2008) propose to record the traces of process implementations within the scope of software quality assurance and to use these records in verification of the software processes. However, this assurance is human-intensive, and it requires competence and experience. Also with this approach, considering that in some situations there is no definition of software process, the chance of using data for process management is low. On the other hand, there are few systematic methods and studies for verification of software process in the literature (Bröckers & Gruhn, 1993), (Holzmann & Smith, 1999), (Yoon, Min, & Bae, 2001), (Silva & Travassos, 2004).

Process mining is a kind of process management technique which is used commonly in business process management in recent years (Holzmann & Smith, 1999). It is an analysis technique between data mining and machine learning at one side, and process modelling at the other side (Aalst, 2011). Considering event logs that are hold on the automation that supports business processes, this technique is implemented for achieving three aims (Aalst, 2011); 1) discovery of process, 2) verification of process implementation according to identified process, 3) enrichment of process by detecting the differences of process implementations.

Process mining techniques are carried out in a limited number of studies for software process (Bröckers & Gruhn, 1993), (Cook & Wolf, 1998), (Cook & Wolf, 1999), (Jensen & Scacchi, 2005), (Kindler & Rubin, 2006), (Rubin & Günther, 2007), (Akman & Demirörs, 2009), (Samalikova & Kusters, 2011). The lack of tool infrastructure supporting the entire software development process and the difficulties in integration of process data from separated tools and assets complicate the use of process mining for software processes. Nevertheless, if software processes are defined event or state based and process flow is supported by a tool, then detection of process by analyzing data, determination of process variations in implementation, or improvement of the process can be possible.

With this opportunity in mind, we first experienced two process mining algorithms, which are widely used in business process management, for software process verification and based on the positive results of this experience, we developed a plug-in that automates the verification using process mining algorithms to an Eclipse-based, open-source EPF Composer (Haumer, 2006) tool. The plug-in was designed to include three basic features: "Process Variants Mining", "Conformance Analysis", and "Statistical Analysis".

Related Content

Using Video Tutorials to Learn Maya 3D for Creative Outcomes: A Case Study in Increasing Student Satisfaction by Reducing Cognitive Load
Theodor Wyeld (2021). *Research Anthology on Recent Trends, Tools, and Implications of Computer Programming (pp. 1706-1742).*
www.irma-international.org/chapter/using-video-tutorials-to-learn-maya-3d-for-creative-outcomes/261098

Service Composition Based Software Solution Design: A Case Study in Automobile Supply Chain
Tong Mo, Jingmin Xu, Zhongjie Wang, Yufei Ma, Heyuan Huang, Yuan Wang, Ying Liu, Jun Zhuand Xiaofei Xu (2012). *Computer Engineering: Concepts, Methodologies, Tools and Applications (pp. 266-277).*
www.irma-international.org/chapter/service-composition-based-software-solution/62447

Text Mining for Analysis of Interviews and Questionnaires
Yasuhiro Yamada, Kanji Katoand Sachio Hirokawa (2011). *Kansei Engineering and Soft Computing: Theory and Practice (pp. 52-68).*
www.irma-international.org/chapter/text-mining-analysis-interviews-questionnaires/46391

Role of Knowledge Workers in Business Process and Innovation
Appasaheb Naikaland Mayank Bapna (2020). *Disruptive Technology: Concepts, Methodologies, Tools, and Applications (pp. 1185-1197).*
www.irma-international.org/chapter/role-of-knowledge-workers-in-business-process-and-innovation/231238

Grouping Concept in Optimum Sizing of Truss Structures: Optimization of Truss Structures
Gebrail Bekda, Sinan Melih Nigdeliand Osman Hürol Türkakn (2018). *Handbook of Research on Predictive Modeling and Optimization Methods in Science and Engineering (pp. 94-120).*
www.irma-international.org/chapter/grouping-concept-in-optimum-sizing-of-truss-structures/206746