Automated Process-Centric Quality Constraints Checking for Quality Assurance in Safety-critical Systems

Christoph Mayr-Dorn[1] Michael Vierhauser[2] Stefan Bichler[3] Felix Keplinger[4] Jane Cleland-Huang[5] Alexander Egyed[6] Thomas Mehofer[7]

Abstract: This abstract summarizes the work published as an ICSE 2021 research track paper [Ma21] available at https://doi.org/10.1109/ICSE43902.2021.00118. We propose an approach that, on the one hand, assists in checking compliance with traceability requirements but, on the other, hand allows engineers to temporarily deviate from the prescribed software engineering process. Through the observation of developer activities in the form of changes to engineering artifacts in tools such as Jira or Jama, we build up a representation of the ongoing process progress. This tracking in the background does not force the software developer to work only on activities as defined in a process description. At the same time, it enables us to provide timely feedback to the developer on whether tasks fulfill all QA criteria. This approach lifts the burden off QA engineers in manually checking QA constraints, often a time-consuming, tedious, and error-prone task where feedback reaches developers usually very late. We evaluate our approach by applying it to two different case studies; one open source community system and a safety-critical system in the air-traffic control domain. Results from the analysis show that trace links are often corrected or completed after the fact and thus timely and automated constraint checking support has significant potential on reducing rework.

Keywords: software engineering process; traceability; developer support; artifact change monitoring

1 Summary

In safety-critical systems, traceability requirements are often mandated by standards (e.g., ED-109A for air traffic management systems) as one form of quality assurance (QA) mechanism. Yet developers and QA engineers are overwhelmed by the complexity and extent of adhering to and evaluating QA constraints. Inspection of fine-grained constraints over engineering artifacts and their traces thus typically occurs only at the end of the engineering work when subsequent feedback often interrupts developers who have moved on to their next task. Such inspection during the engineering process is difficult as the current industry practice is to use semi-formal process descriptions that are not executable.

[1] Johannes Kepler University, Linz, Austria christoph.mayr-dorn@jku.at
[2] Johannes Kepler University, Linz, Austria michael.vierhauser@jku.at
[3] Johannes Kepler University, Linz, Austria
[4] Johannes Kepler University, Linz, Austria
[5] University of Notre Dame, Notre Dame, USA JaneHuang@nd.edu
[6] Johannes Kepler University, Linz, Austria alexander.egyed@jku.at
[7] Frequentis AG, Vienna, Austria thomas.mehofer@frequentis.com

To this end, we propose our Passive **Pro**cess Execution and Quality **Con**straint Support Framework **ProCon**. Two key aspects that characterize our framework are (a) the integrated handling of explicitly distinct processes and constraints, and (b) the tracking of engineering progress achieved through linking process specification to software engineering artifact details. The main difference to contemporary process support solutions [Gr02] is the framework's ability to track the process in the background, based on the software engineers' activities performed in the tools they are using in their daily work rather than requiring engineers to interact with a process engine. Hence, engineers are free to deviate from the process, but still receive guidance even in the presence of a deviation.

Deviations from the process are tracked via process and quality constraints evaluated in the Drools rule engine. These constraints, and their respective evaluation results, are treated as first-class citizens in the software engineering process for determining process progress and serving as explicit feedback to the developers via the framework's web-based user interface.

We implemented our framework as a Java prototype connecting to Jama and Jira as representative engineering tools that enable managing of requirements, design documents, work items, test cases, etc as well as the traces amongst these artifacts. We evaluated our approach by extracting the change history of the artifacts involved in processes from the open-source UAV project Dronology [CHVB18] and our industry partner Frequentis, a supplier of air traffic management systems. For the industry use case, we tracked the process deviations and constraint violations during the replay of more than 14,000 artifact changes for 109 process instances. We found **temporary** deviations and violations in 20% of processes, concluding that automated developer feedback is highly desirable.

1.1 Data Availability

The prototype and data used in the original paper are available at Figshare `https://doi.org/10.6084/m9.figshare.12840053`.

Literaturverzeichnis

[CHVB18] Cleland-Huang, Jane; Vierhauser, Michael; Bayley, Sean: Dronology: An Incubator for Cyber-Physical Systems Research. In: Proc. of the 40th Int'l Conf. on Software Engineering: New Ideas and Emerging Results. ICSE-NIER '18. ACM, S. 109–112, 2018.

[Gr02] Gruhn, Volker: Process-centered software engineering environments, a brief history and future challenges. Annals of Software Engineering, 14(1-4):363–382, 2002.

[Ma21] Mayr-Dorn, Christoph; Vierhauser, Michael; Bichler, Stefan; Keplinger, Felix; Cleland-Huang, Jane; Egyed, Alexander; Mehofer, Thomas: Supporting Quality Assurance with Automated Process-Centric Quality Constraints Checking. In: 43rd IEEE/ACM International Conference on Software Engineering, ICSE 2021, Madrid, Spain, 22-30 May 2021. IEEE, S. 1298–1310, 2021.