Towards Alignment of Processes, Tools, and Products in Automotive Software Systems Development

Joachim Schramm
and Andreas Rausch
and Daniel Fiebig

Clausthal University of Technology
Department of Computer Science &
Institute for Applied Software Systems Engineering
Goslar
Germany
Email: forename.surname@tu-clausthal.de

Oscar Slotosch
and Mohammad Abu-Alqumsan

Validas AG
Munich
Germany
Email: surname@validas.de

***Abstract*—Numerous quality standards impact the lifecycle of software and system development in the automotive industry. Hereby, quality is evaluated through rigorous assessment of the deployed processes, tools and products. Yet, although these three aspects go hand-in-hand, they are typically assessed separately and manually. Moreover, system providers are increasingly challenged by media breaks coming along with the necessity to integrate processes and tools, and to facilitate data exchange among these tools. Consequently, this adds more demands and challenges on certification. This paper presents the TOPWATER approach, whereby a unified metamodel is used to specify how processes, standards and tools are linked. We introduce TOPWATER from the conceptual as well as from the technical perspective. Shown also in particular is the integration of the approach into the Validas' qualification methodological framework, which is used for qualification and classification of toolchains; a requirement of safety standards imposed on all tools used to develop safety-related products/items. This example from the Automotive Software Engineering field is used for evaluation, where outputs from the TOPWATER method supported Validas process assessment by TÜV. In this pretest, TOPWATER has demonstrated its feasibility.**

Keywords–Process-Tool Integration; Toolchain; Software Process; Standards; Certification.

I. INTRODUCTION

Today's software engineering of automotive software systems is impacted by a number of safety standards, such as ISO 26262 and IEC 61508. Hereby, quality and safety are assured through rigorous assessments of the processes and tools used, and the final product. For instance, process quality is achieved by formulating and assessing requirements, e.g., test procedures or code coverage levels. Likewise, tools and particularly critical tools must be classified [1]. Finally, the product's quality is evaluated by checking, e.g., if the product development followed a defined process and process steps and tools involved were developed and used according to the respective standard's requirements. However, even though these three aspects are highly interrelated, they are mostly implemented and assessed individually and manually. This can lead to issues regarding consistency and completeness, e.g., a process that was not implemented as defined, or use of tools not appropriately qualified. Currently, available assessment tools mostly cover these aspects in an isolated manner, such as process modeling (e.g., Eclipse Process Framework) or toolchain analysis (e.g., Validas Toolchain Analyzer). If at all, integration of these aspects is done manually using Microsoft Excel.

A. *Problem Statement & Objective*

In certification of automotive software systems/items/products, tools, processes, and products need to comply with the relevant standards. Yet, most process-tool ecosystems are loosely coupled thus challenging certification. The approach presented in this paper aims at providing a solution that allows for seamlessly integrating software system development processes and the tools used to perform, enact and track these processes. Our approach particularly aims at closing content-related and semantic gaps that hinder a seamless integration by monitoring across-tool artifacts and products of the different processes.

B. *Contribution*

In this paper, we present the TOPWATER approach to support a seamless integration of development processes and tools. Our approach is grounded in a metamodel, which allows for linking software process models and development tools. Notably, to support automotive software systems development, our approach is integrated into the Validas' qualification methodological framework, which is used for qualification and classification of toolchains. We present our approach as a conceptual model, which is translated into a technical metamodel (based on the Eclipse Modeling Framework), and we show the implementation of our approach in the toolchain of Validas AG that is primarily used in Automotive Software Engineering projects.

C. *Outline*

The remainder of the paper is organized as follows: Section II presents a short background supported by some practical observations (Subsection II-A) and further provides a short review of selected related work (Subsection II-B). Section III presents the approach from a conceptual and technical perspective (Subsection III-A), shows the feasibility based on a TÜV pretest (Subsection III-B), and critically discusses the results achieved so far (Subsection III-C). We conclude the paper in Sect. IV, including the limitations (Subsection IV-A) and the future work (Subsection IV-B).

II. Background & Related Work

This section provides the background and the motivation of our research using observations from practice. Furthermore, we present work related to our research.

A. Observation from Practice

Aligning processes and tools in automotive product development is challenging, since heterogeneous ecosystems introduce media breaks and a multitude of data formats. As such, this puts a strain on the product development process as a whole since consistency and completeness, e.g., of models and software, must be ensured as a prerequisite for certification. Aiming at gathering some field information with respect to process-tool alignment, the authors have interviewed nine practitioners from six different divisions within a large German car manufacturer. These interviews were part of a joint automotive process-development workshop. Table I lists the questions which were asked and a summary of the respective answers. Yet, for confidentiality reasons, we can only provide summaries of the interview findings. All information about the respective persons, company and project contexts were removed prior to extracting the information presented hereby. In a nutshell, the results highlight two main factors for causing process- and tool-chain breaks: technical (e.g., isolated incompatible tools) and Human factors (e.g., too extensive and/or complicated processes are ignored, and complex tools pose great challenges to users).

TABLE I. Summary of findings from interviews with six divisions of a large German car manufacturer on experiences concerning process- and tool-chain breaks.

Question	*Summary of Findings*
What do you consider a tool-chain break?	Practitioners consider an employee that is acting as a "man in the middle", i.e., who transfers data manually from one tool to another, a tool-chain break. Another potential break is seen in situations that disturb the normal workfow, e.g., where employees try to familiarize themselves with a new tool that has replaced a previously used one. In addition to these human factors, communication and data transfer issues constitute tool-chain breaks. For example, an integration of DOORS and JIRA, or Vector Software's PREEvision and Excel can—if at all—convert data from tool A to B, but not vice versa.
What do you consider process-chain break?	Practitioners consider potential discrepancies between defined (standard) processes and their implementation and/or enactment as breaks in the process-chain. Particularly, undocumented process deviations might impact compliance checks that take place at subsequent stages, and therefore are considered as a source of risk.
What is the main source for such breaks?	Practitioners point out two major sources for breaks, of which the Human stands as the first and obvious one. Humans might be overwhelmed by documentations, do face the need to interpret process descriptions, and/or do struggle with the complexity of the tools at hand. All this may lead to sloppiness in the process implementation/enactment as well as to going for shortcuts, notably in projects under time pressure. The second major source is concerned with technology. For instance, different tools assembled in a tool-chain might cause data exchange complications. Also, expert tools, which are typically used by a restricted number of persons, can cause breaks if many other employees want to consume or provide data, which these tools respectively generate or consume.
Do you have support available or counter-measures defined to deal with such breaks?	The interview revealed contradicting answers by the different divisions. Three divisions argue for working according to the book, i.e., an employee has to read the process description before performing activities/producing artifacts, and involved people should not (only) rely on other 'experienced' people that explain their ways of work. On the other hand, the other three divisions opt for the exact opposite way. In general, the interview participants name external consultants as a countermeasure to avoid process- and tool-chain breaks. At the tool-level, participants express their intentions to intensify tool evaluation towards bidirectionally compatible tools to avoid issues with data transfer/exchange.

B. Related Work

What basically renders the alignment of process- and tool-chains challenging is that it involves different problem fields, such as data exchange formats, process integration, workflow management, and process enactment. Moreover, in highly regulated domains, such as automotive, integrated process-toolchains, which are used as a part of the product development process, usually require a certification.

To the best of our knowledge, the approach presented in this paper is unique, yet it is built upon concepts and solution approaches for different problems. A major issue addressed by TOPWATER is the systematic design of a toolchain in which different tools are assembled to create an integrated work environment. Among other things, the variety of tools available challenges companies across the Globe. For instance, Portillo-Rodríguez et al. [2] provide an overview of tools used in globally distributed software development. They classified 132 different tools, yet mentioned that only a small percentage is practically relevant.

Different platform providers offer solutions to consolidate the "tool zoo". In particular, two approaches can be identified: (i) open platforms like the Eclipse platform and (ii) commercial tools/toolchains such as Microsoft's Team Foundation Server or Rational's Team Concert. The latter approach usually provides interfaces that can be used by third-party developers. These platforms remain, however, closed to a certain extent and additionally impose high expenditure and infrastructure requirements. On the other hand, open platforms like Swordfish [3], agosense.symphony [4], ToolNet [5], or ModelBus [6] provide mechanisms for technical integration only. These platforms require specific adapters for the tools to be integrated—in the worst case, data is transferred as plain XML. Nevertheless, both open and commercial platforms have in common that adapters have dependencies to the respective platforms, which requires effort to keep them up to date and operational. TOPWATER aims at addressing this issue by providing a model-based mapping approach. A model connects the different tools and, in the best case, allows for generating the required connections. From this perspective, the approach followed in TOPWATER is comparable with the SPRINT platform [7]. SPRINT aims at providing a unified view on the different components of complex systems developed in an interdisciplinary manner. It connects the different domain-specific expert tools by providing means to design and transform data to be exchanged among the different tools. The actual connection is realized using OSLC [8], which connects development artifacts and adds semantics to these artifacts and their relationships.

Aiming at facilitating process modeling and enactment, TOPWATER is built on several concepts. Other than business processes, software development processes are hard to enact. For instance, Rozinat and van der Aalst [9] show how conformance checking for business processes is realized using actual behavior. Furthermore, in his keynote, van der Aalst [10] presents the current state of (business) process mining. However, different from business processes, development processes are way less predictable (e.g., due to developers' creativity, situation-specific problem solutions, changing requirements). Nevertheless, different approaches exist to prepare and implement process enactment. Similar to the TOPWATER approach, transformation of process models to allow for process execution or document generation is quite common [11]. Furthermore, rule-based execution of processes using explicit modeling languages is a well-established concept, e.g., [12]. However, approaches proposed so far primarily address the conversion of process models into a format that allows for automating their respective processes. Integrating the processes into a complex conglomerate of processes, products and tools that have to be certified, so far, received little attention. In order to support the aforementioned situation, checking process properties is key (see also Cobleigh et al. [13]). TOPWATER addresses these requirements by utilizing a metamodel to integrate the different aspects and, moreover, to allow for formally evaluating consistency, compliance, and so forth. In particular, TOPWATER helps addressing the determination of the so-called *Tool Confidence Levels* (TCL; as defined by ISO 26262 [14]). An assessment is performed automatically using formal methods. Further details on the tool classification and qualification approach for ISO 26262 can be taken from Conrad et al. [15] in which authors share experiences concerning implementing an approach for development and verification tools. Given by the context, TOPWATER so far supports Validas' *Tool Chain Analyzer* (TCA; [1]) [16]. However, TOPWATER is designed with flexibility in mind that other tools, such as *RapiCover Aero* or *RapiTime Aero* can be targeted [17]. These tools help assessing the compliance with the DO-178B or DO-178C requirements, which are relevant for aerospace applications.

III. The TOPWATER Approach

We present our approach to support developing seamlessly integrated processes and toolchains. In Sect. III-A, we present the approach at conceptual and technical levels. Section III-B presents the validation, and Sect. III-C critically discusses our approach.

A. Solution Approach

The TOPWATER approach comprises two parts: a conceptual generic solution and a technical approach for direct implementation in the context of particular toolchains. In the following, we introduce both parts with more emphasis on the technical solution and its concrete implementation.

1) Conceptual Model: TOPWATER is a collection of metamodels and tools, which are illustrated in Fig. 1. In its core, TOPWATER allows for

- modeling software development processes (Fig. 1; *Process Layer*),
- modeling tools as part of a toolchain (Fig. 1; *Tool Layer*), and
- modeling use-case-based mappings between processes and tools. (Fig. 1; *Mapping Layer*).

Following the modeling of tool- and process-chains step, three major steps may take place. These are represented with partitions and shown in Fig. 1 as vertical "swimlanes".

- validation step: in which process engineers validate the different interconnected models before a project starts.
- operation step: in which different concrete process artifacts (or artifact instances) and project results are created.
- controlling step: project managers use the generated models to implement a quality-gate-based controlling to check the fulfillment of the process- and tool-related requirements, and to check whether such requirements are likely to be fulfilled in future project stages or not.

Grounding TOPWATER in a metamodel provides manifold options to support the models' quality assessment—at design time and during the different project phases alike.

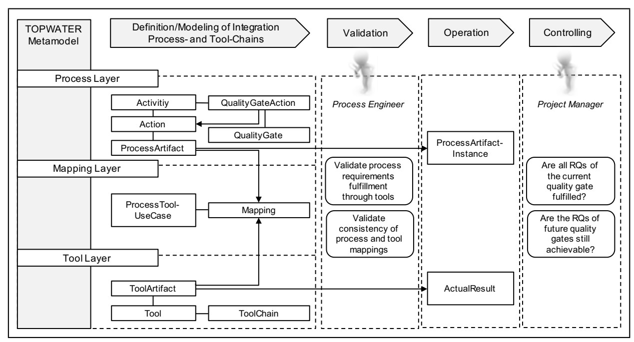

Figure 1. Overview of the conceptual model of the TOPWATER approach including the different metamodel layers, the project phases and selected model elements and project-related activities.

2) Implementation: As a proof of concept, TOPWATER is integrated with the Validas toolchain, which is usually used for formal tool qualification [1], i.e., a tool can be qualified according to safety standard requirements and a tool can be mapped to an assessable process to create a certification-ready work environment. To allow for easy integration in different toolchains, TOPWATER's metamodel is developed in the Eclipse Modeling Framework (EMF), which additionally allows for generating a modeling tool from the metamodel. Figure 2 provides an overview of the key elements in the metamodel, of which certain data types for standards or requirement classes are used to manifest the transition from the generic concept (Fig. 1) to the actual practical implementation.

From the technical perspective, a TOPWATER project comprises `Components`, `Processes`, and `ToolChains`. Products and components, in particular, are mapped to `Requirements`, e.g., concerning the targeted standard or required safety integrity levels by the tool qualification. `Products` together with `Actions`, define a `Process`. The actions allow for designing the process to be checked. Process modeling follows the UML principles (including, e.g., start-, split-, and conditional actions) and, hence, users can model object and control flows alike. For instance, products can be

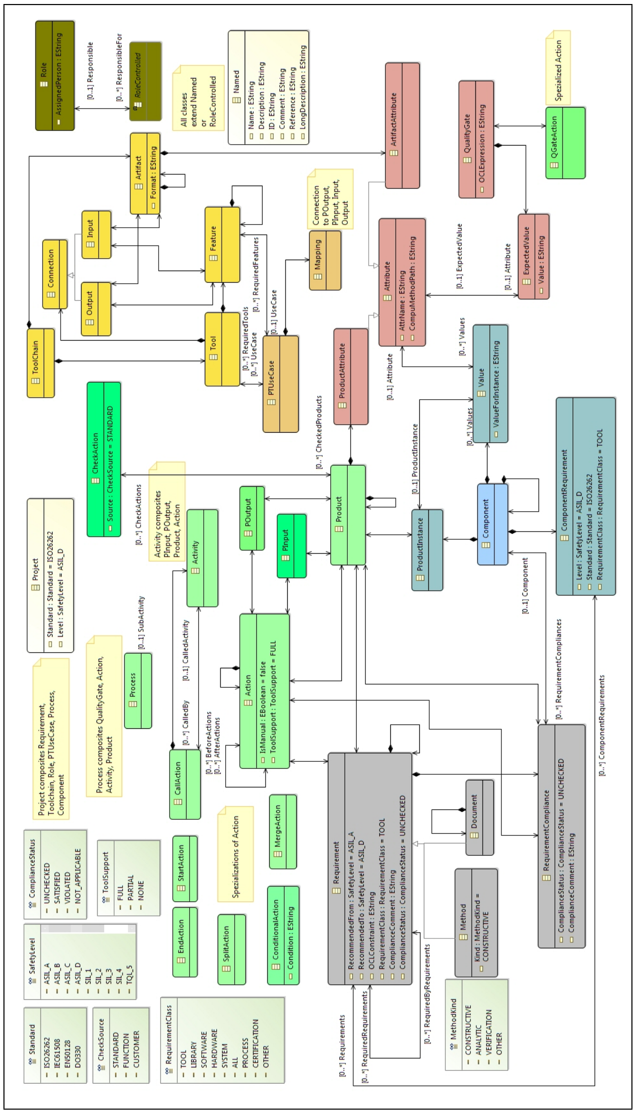

Figure 2. Key elements of the TOPWATER metamodel, developed in the Eclipse Modeling Framework (EMF) to allow for easy integration with existing company toolchains.

inputs (`PInput`) and outputs (`POutput`) of activities. Similarly, toolchains contain `Tools`, `Artifacts`, `Features`, and relations, e.g., artifacts are inputs/outputs of tools.

A key feature of the TOPWATER tool is the ability to provide mappings. A mapping serves the requirement to be able to link tools and processes, such that the development process can be tracked and, thus, aids the different certification requirements. Mappings are part of the so-called `PTUseCases`, which link process inputs and tool outputs (and vice versa), i.e., process steps are connected to tools that realize them.

The metamodel allows for several validation operations and quality assurance activities, which in turn can aid certification needs. TOPWATER uses a validation model based on quality gates. A `QGateAction` is a specialized action, which has one or more `QualityGates` assigned. By design, a quality gate is a predicate expressed in OCL to define product states and product attribute values at certain project stages. Quality gates represent critical steps in the evolution of artifacts and, within this evolution, several product-specific expressions must be true for the project to continue.

For the actual validation in the context of a quality gate, TOPWATER provides two views, to which we refer as

- *closed world view* (CWV)
- *open world view* (OWV).

The CWV focuses on supporting completeness checks of artifacts. To this end, an artifact-specific expression is assumed `false` (i.e., a quality criterion is not fulfilled yet) whenever it is not evaluated as `true` for a certainty. Thus, performing a validation on a quality gate and all products assigned to it provides an overview of all incomplete artifacts and, particularly, the attributes that cause the tests to fail. On the other hand, the OWV assumes an expression to be `true`, regardless of its actual evaluation. Consequently, this view provides a broader picture of the modeled project and, to a certain extent, supports prediction. This is achieved by checking which subsequent quality gates might be fulfilled given the current project status. The prediction allows for identifying quality gates that can be reached according to plans, as well as identifying those quality gates that cannot be reached anymore, should the process be executed as planned and without modification. Such predictions can be used for early deviation detection, and therefore enable project managers to initiate counter actions as early as possible.

3) Integration into the Validas Environment: Figure 3 provides an overview of the compliance methodology of the Validas AG. The model-based *Validas Qualification Methodology* (VQM) is used for classifying and qualifying tools and toolchains according to safety standards. The VQM is informally described with an informal process description. Central to the VQM is the *Tool Chain Analyzer* (TCA), which is a modeling tool that is developed by Validas AG and is used to develop the so-called *Qualification Support Tools* (QSTs). QSTs are shipped to customers, so that qualification can be performed within the customer environment to automatically generate work products relevant to the target safety standards, such as the Tool Qualification and Tool Criteria Evaluation Reports in ISO 26262. Evidence is generated for each developed QST to assure that (i) procedures are compliant with safety standards, and (ii) procedures were adhered to. The former is typically covered by a compliance report (CR), whereas the later is typically done by performing *Verification & Validation* (V&V) and generating the V&V report. The QST and the quality assurance documents (i.e., CR and V&V report) are altogether referred to as *Qualification Kit* (QKit).

Figure 3 additionally shows the TOPWATER's integration into the otherwise isolated steps of the VQM. As a prerequisite, however, the VQM process is formally modeled using *AutoFocus3* (AF3; [18]). AF3 is an open source modeling tool that allows for model-based development of embedded systems, with support for formal modeling and rigorous quality assurance of the models (including, e.g., validation, verification, model checking and simulation). By using AF3 for modeling the processes of the VQM, these processes can be formalized and checked using the quality assurance mechanisms of the tool. The formally modeled and checked VQM processes can be easily imported into the TOPWATER tool. The TOPWATER tool additionally receives the targeted standard(s) as input. The

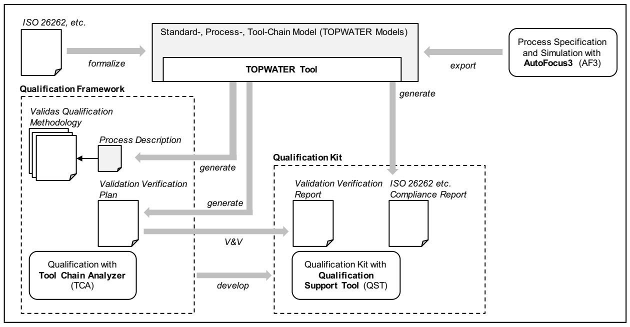

Figure 3. Overview of the Validas compliance methodology.

target standard(s) is/are used to create the mappings of the different processes and tools (which happens here to be the tool chain modeling tools) (see Fig. 2). Using the different standard-, process-, and tool models, the TOPWATER tool can automatically generate: (i) the compliance report of the developed/modeled processes (ii) the V&V plan as a checklist, according to which V&V can be performed and V&V reports can be generated (iii) a formal description of the processes, which when combined with the informal description is used to guide involved people in creating QKit projects.

B. Validation by Example

This section presents a validation of the TOPWATER tools by example. The validation presented hereby is based on an audit performed by TÜV (TÜV = Technischer Überwachungsverein, engl: Technical Inspection Association); a certification body that provides independent inspection and product certification services (in Germany and worldwide). The audit aimed at evaluating if the generated files from TOPWATER are adequate to support process/product certification. In this pretest the TOPWATER approach was preliminarily evaluated for certification readiness. At the TOPWATER tool side, this involved:

- Modeling of the tool qualification process
- Mapping of standards' requirements and their mapping to tools
- Mapping of processes and tools
- Generation of compliance reports and V&V plan

Figure 4 shows an excerpt of the VQM process modeled in AF3 in which the qualification component of the process is shown. This component takes a QKit and the tool (model) to qualify and executes the qualification process. The component shown in Fig. 4 contains the three actions `classify`, `plan`, and `validate`. The `validate` action creates the *Tool Qualification Report* (TQR; to be delivered to the safety manager for review) and the *Tool Safety Manual* (TSM; to be delivered to the tool user). The result of importing the VQM AF3 model into TOPWATER is shown in Fig. 5 showing the manifestation of the different metamodel elements from Fig. 2. Specifically, the figure shows the aforementioned actions `classify`, `plan` and `validate`, and the products QKit, TSM, TQR, TCS, and TQP. Also, the input and output relations are shown that illustrate the workflow. For example, the element `POutput Validate_to_TSM` connects the

Figure 4. Excerpt of the VQM modeled in AF3 to be exported to TOPWATER.

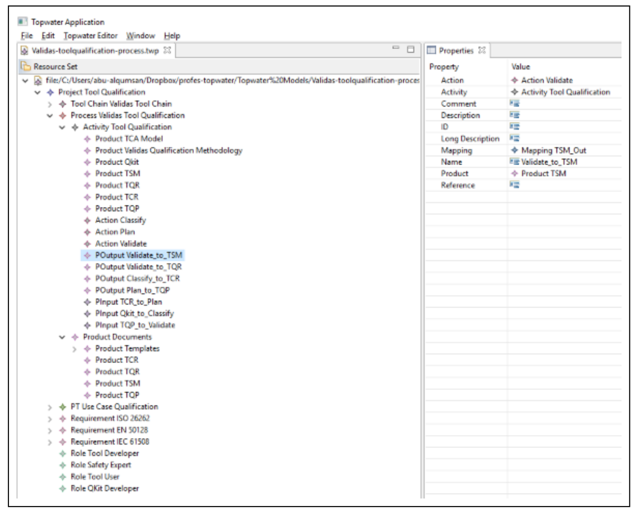

Figure 5. The TOPWATER tool after importing the VQM from AF3 (node *Validas Tool Qualification*, which includes the activities and products involved in a qualification).

action `validate` and the corresponding output product TSM.

Providing the formalized VQM process to TOPWATER allows for integrating all tools involved in the tool qualification process into one framework. Among other things, Fig. 5 shows *Requirement* nodes, which contain models of the target standards for which the compliance must be checked. Figure 6 shows a part of the ISO 26262 [14] standard used for the TÜV audit mentioned at the beginning of this section. The selected item *8-11.4.6 Qualification of a software tool* demonstrates

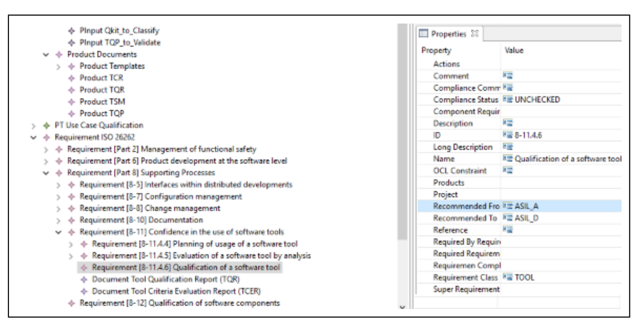

Figure 6. Expanded nodes for tools and requirements in the TOPWATER tool.

how requirements are modeled within TOPWATER. It shows, inter alia, that requirements have compliance status, requirement class, and recommendations concerning their application with respect to the different ASIL levels—in this particular case, the selected requirement must be checked for all levels (i.e., from ASIL A to ASIL D).

To carry out the actual tool qualification as required by standards, it is necessary to explicitly state which tool is used to perform a specific process step (including the specification of the artifacts involved). For this, the TOPWATER metamodel (Fig. 2) defines the class `PTUseCase`, which handles the mapping of processes, products, and tools. Figure 7 shows the expanded node *Qualification* that realizes this mapping. In particular, the *Qualification* use case maps process inputs to tool/toolchain inputs and, respectively, process outputs to tool/toolchain outputs. For example, the mapping *TQP_Out* connects the `POutput` *Plan_to_TQP* from the validation process with the output *TQP Output* from the toolchain. The metamodel ensures that only valid mappings can be made, e.g., `POutput` elements can only be mapped to other `POutput` elements, which is ensured by a just-in-time filtering and type validation.

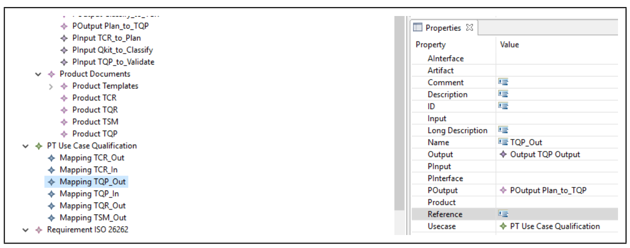

Figure 7. Expanded nodes for PTUseCases and the process and tool mappings.

Beyond the type-based standard validation, the TOPWATER tool also checks the models for completeness and inconsistencies. The platform allows for defining required rules, i.g.: check if all requirements have names and descriptions provided, and if the provided names are unique.

The previously shown steps give an overview of the activities necessary to develop the different models for processes, products and tools, and how to connect them to modeled standards for which the compliance must be checked. The tool further helps performing the required activities to a large extent and allows for documenting the mappings appropriately, i.e. the V&V-related documents. Here, the *V&V Plan* contains a description of the V&V strategy, which, inter alia, includes the supported safety standards, the tool adequacy, and a description of the different checks for standard compliance or the components involved. Complementing, the *V&V Report* contains the check results. Notably the *V&V Report* shows the sheer mass of documentation that needs to be delivered for a certification process thus underlining the benefits a model-based approach offers.

C. Discussion

The previous sections have presented the TOPWATER approach and its first implementation within the Validas processes and tools. This section discusses the benifits and the limitations of the current state of TOPWATER. As a first step, we highlight in Table II the extent, in how far TOPWATER addresses the challenges mentioned by practitioners, which were summaries in Table I.

TABLE II. Discussion of the current state of the TOPWATER approach following the structure of Table I.

Question	*Summary of Findings*
What do you consider a tool-chain break?	Practitioners consider project employees or "man in the middle" and data exchange/transfer as major issues. In the current implementation, TOPWATER does not address tools with automation interfaces. Yet, TOPWATER makes a first step by providing an integrated and uniform model that allows for connecting the different parts. Automation, however, remains subject for future works.
What do you consider process-chain break?	Practitioners consider discrepancies between "as-planned" and "as-is" processes as a major problem. TOPWATER addresses this aspect through its manifold process modeling and model checking options including options that check compliance against the targetted standards, such as ISO 26262 or DO 330.
What is the main source for such breaks?	Practitioners consider human factors and technological aspects as major sources for breaks. TOPWATER partially addresses these problems. The human factor is addressed by the manifold process modeling and checking capabilities, e.g., importer from Microsoft Visio or AF3, and process enactment with quality gates based on process constraints. The technological aspects are initially covered by the ability of TOPWATER to model tools and toolchains. Given that the respective tools provide sufficient options for data exchange, this exchange can be modeled accordingly and integrated with the process-, product- and tool mapping.
Do you have support available or counter-measures defined to deal with such breaks?	See previous comment. Furthermore, the TOPWATER tool provides users with comprehensive guidance and different quality checks to guard against possible breaks, e.g., type-based constraints or more comprehensive validity checks.

The current implementation of the TOPWATER approach has been run through a TÜV pretest and demonstrated its feasibility. Furthermore, the current implementation solves several of the challenges identified (Table II). Specifically, the problem of proving that a defined process has been performed is addressed. The TOPWATER tool allows for importing process models, i.e., the development process of the companies running a development project targeting an automotive application. The imported process is connected with the tools used to enact the process and the standard requiring certain activities executed to be eligible for certification. The mapping provided by TOPWATER allows for creating an integrated model that can be evaluated for consistency and compliance and, furthermore, supports the generation of reports as requested by the certification authorities. Moreover, the TOPWATER tool can also be integrated with the project as such. By continuously maintaining the state of the process through constraint-based quality gates, project progress can be tracked and, to a certain extent, predicted such that early plan deviation is possible. Finally, the qualification method implemented using TOPWATER is able to validate itself. Using AF3 as process modeling tool, the qualification process is formalized thus allowing for formal validation of the process (which is also possible for any other method that is incoming via the AF3 interface). Nevertheless, several practical requirements are not (yet) implemented, which we further discuss in the Sect. IV-B.

IV. Conclusion and Future Work

The present paper has presented TOPWATER; a methodological approach that allows for joint and integrated modeling of processes, products, and tools. The objective of such modeling is to reduce breaks in process- and toolchains, and hence to reduce friction in the development processes. In industries, where such breaks may have a high impact on efficiency, like automotive, the proposed approach might be of great interest. The integrated approach helps companies to ease and accelerate certification (and recertification) efforts. Specifically, we have shown the integration of TOPWATER into Validas AG qualification methodology that supports the classification and qualification of tools and toolchains, a key activity required by automotive safety standards. An initial evaluation of the presented approach was carried out, where the TOPWATER tool was exploited to automatically generate documents that supported Validas AG process assessment by TÜV. This evaluation has successfully demonstrated the feasibility of the approach.

A. Limitations

Within the current state of the TOPWATER tool, several standards are already modeled or still under development. Furthermore, different converters/adapters are implemented, e.g., to integrate formal process models generated by other modeling tools. However, the TOPWATER approach and the respective tools still have some limitations (Table II). For instance, despite the fact that the modeling approach is per se accurate and correct, the modeling activities, in its current form, remain a complex activity that requires expert knowledge in the field. Moreover, several practical problems identified throughout the project are currently only partially addressed by the tool, i.e., the concept is developed, but the tool does not deliver its features fully. For example, several aspects that are part of the automation engine still require manual work and checks.

B. Future Work

Future work comprises the completion of the automation engine to implement and, eventually, provide all TOPWATER features to its users. In the present work, we have also shown the feasibility of the approach for the tool qualification methodology. Hereby, however, the scope of the safety standards which is covered by projects of this nature is relatively limited. For Future work, evaluation with other applications, with large number of tools and large scope of targeted standards are planned.

References

[1] M. Wildmoser, J. Philipps, and O. Slotosch, "Determining potential errors in tool chains: Strategies to reach tool confidence according to iso 26262," in Proceedings of the 31st International Conference on Computer Safety, Reliability, and Security, ser. SAFECOMP'12. Berlin, Heidelberg: Springer-Verlag, 2012, pp. 317–327.

[2] J. Portillo-Rodríguez, A. Vizcaíno, M. Piattini, and S. Beecham, "Tools used in global software engineering: A systematic mapping review," Information & Software Technology, vol. 54, no. 7, 2012, pp. 663–685. [Online]. Available: https://doi.org/10.1016/j.infsof.2012.02.006

[3] [Online]. Available: http://www.eclipse.org/swordfish/

[4] [Online]. Available: http://www.agosense.com/agosense.symphony

[5] [Online]. Available: http://www.es.tu-darmstadt.de/forschung/overview/

[6] [Online]. Available: http://www.modelbus.org/en/modelbusoverview.html

[7] [Online]. Available: http://www.sprint-iot.eu/

[8] M. Saadatmand and A. Bucaioni, "Oslc tool integration and systems engineering – the relationship between the two worlds," in Proceedings of the 2014 40th EUROMICRO Conference on Software Engineering and Advanced Applications, ser. SEAA '14. Washington, DC, USA: IEEE Computer Society, 2014, pp. 93–101. [Online]. Available: http://dx.doi.org/10.1109/SEAA.2014.64

[9] A. Rozinat and W. M. P. van der Aalst, "Conformance checking of processes based on monitoring real behavior," Inf. Syst., vol. 33, no. 1, Mar. 2008, pp. 64–95. [Online]. Available: http://dx.doi.org/10.1016/j.is.2007.07.001

[10] W. v. d. Aalst, "Big software on the run: In vivo software analytics based on process mining (keynote)," in Proceedings of the 2015 International Conference on Software and System Process, ser. ICSSP 2015. New York, NY, USA: ACM, 2015, pp. 1–5. [Online]. Available: http://doi.acm.org/10.1145/2785592.2785593

[11] M. Kuhrmann, G. Kalus, and M. Then, "The process enactment tool framework-transformation of software process models to prepare enactment," Sci. Comput. Program., vol. 79, Jan. 2014, pp. 172–188. [Online]. Available: http://dx.doi.org/10.1016/j.scico.2012.03.007

[12] G. E. Kaiser, N. S. Barghouti, and M. H. Sokolsky, "Preliminary experience with process modeling in the marvel software development environment kernel," in Twenty-Third Annual Hawaii International Conference on System Sciences, vol. ii, Jan 1990, pp. 131–140 vol.2.

[13] J. M. Cobleigh, L. A. Clark, and L. J. Osterweil, "Verifying properties of process definitions," SIGSOFT Softw. Eng. Notes, vol. 25, no. 5, Aug. 2000, pp. 96–101. [Online]. Available: http://doi.acm.org/10.1145/347636.348876

[14] ISO TC22/SC3/WG16, "ISO/IEC 26262:2011: Road vehicles – Functional safety," International Organization for Standardization, Tech. Rep., 2011.

[15] M. Conrad, G. Sandmann, and P. Munier, "Software tool qualification according to iso 26262," 2011.

[16] [Online]. Available: http://www.validas.de/TCA.html

[17] [Online]. Available: www.rapitasystems.com

[18] Fortiss, "AutoFocus3 Modeling Platform, v2.11," Online: https://af3.fortiss.org, Februrary 2017.