

# Managing Traceability Information Models: Not Such a Simple Task After All?

Salome Maro, Jan-Philipp Steghöfer, Eric Knauss, Jennifer Horkoff, and Rashidah Kasauli, Chalmers University of Gothenburg

Rebekka Wohlrab, Systemite AB, Gothenburg, and Chalmers University of Technology, Gothenburg

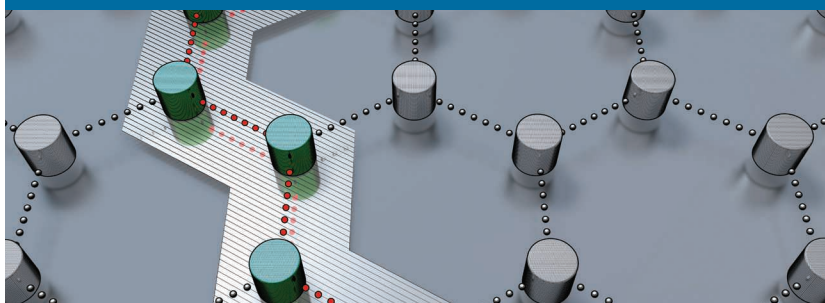
Jesper Lysemose Korsgaard, Florian Wartenberg, and Niels Jørgen Strøm, Grundfos Holding A/S

Ruben Alexandersson, Volvo Cars

*// Practitioners are poorly supported by the scientific literature when managing traceability information models (TIMs), which capture*

*the structure and semantics of trace links. We present our findings from an in-depth focus group about TIM management with three different systems engineering companies. //*

**A CRUCIAL BUILDING** block in a successful traceability strategy for any organization is a well-defined and goal-oriented traceability information model (TIM). A TIM defines the relationships between the diverse artifacts created during system development, often by different teams and parts of an organization. For these relationships, it defines structural aspects, such as trace link types and cardinalities. Additionally, a TIM's semantic information can be informed by organizational structures, information about the workflow, the interaction between products and platforms, and the collaboration between different parts of an organization. This makes a TIM a crucial artifact in an organization and, at the same time, very difficult to change once established. Thus, the main question when defining a TIM is how to capture these aspects and translate them into a model that the individual engineer can successfully work with and use to create value. TIM management includes activities such as defining which relationship types are needed, implementing the designed TIM in a tool, and managing its evolution. Because of the complexity of what a TIM captures and its importance for traceability, each company, and sometimes each project, requires a tailored TIM.



©SHUTTERSTOCK/JNT VISUAL

Digital Object Identifier 10.1109/MS.2020.3020651  
Date of current version: 20 August 2021

Due to this specificity, practitioners lack concrete guidelines when creating TIMs, especially in large systems engineering companies. To address this issue, we conducted a workshop with three companies developing systems from three different domains to understand their current and emerging TIMs and the challenges they face when managing them. The workshop revealed that each of the three companies manage their TIMs in radically different ways but that they face common challenges that are yet to be addressed in the literature. We summarize our findings in this article and describe research challenges that result from them as a call to arms for requirements engineering (RE) and software and systems traceability community.

## Background on TIM Management

The literature on TIM management is split into three main themes: research proposing TIMs for specific purposes, e.g., TIMs for safety certification<sup>1</sup> or for managing non-functional requirements;<sup>2</sup> research proposing generic TIMs that can be reused, e.g., Espinoza et al.;<sup>3</sup> and, finally, tools that support concrete implementation of TIMs that have already been designed, such as Drivalos et al.<sup>4</sup> (For an overview of terms used, see “Important Terms.”) Regardless of theme, however, many of the studies are abstract (e.g., Ramesh and Jarke<sup>5</sup>) or were conducted using small examples that are isolated from realistic company environments. Thus, their generalizability is dubious. In both the research

community and in industry, it is agreed that no single TIM fits every development context; therefore, the number of studies proposing generic TIMs has decreased over time, and research is focused on project-specific or task-specific TIMs.<sup>6</sup>

While it is true that no single TIM fits every development context, concrete guidelines are needed to support practitioners to define, implement, and maintain their specific TIMs. This part of research is lacking, and practitioners have to manage TIMs with little support from the scientific literature. A few studies exist, e.g., the work by Ramesh and Jarke,<sup>5</sup> which provides some reference TIMs that practitioners could use as a starting point for designing their own TIMs. The article discusses issues encountered

## IMPORTANT TERMS

*Traceability* is defined as the ability to interrelate any uniquely identifiable software engineering artifacts to any other, maintain required links over time, and use the resulting network to answer questions of both the software product and its development process.<sup>S1</sup>

A traceability strategy is a plan of action leveraged to design a traceability solution, consisting of a traceability process, tooling, and a TIM.

A TIM defines all of the artifact types that need to be traced and the different relationship types between the artifact types. Figure S1 displays an example of a simple TIM.

### Reference

[S1] Center of Excellence for Software Traceability (CoEST). Accessed: Apr. 29, 2020. [Online]. Available: <http://www.coest.org>



**FIGURE S1.** A simple TIM as an example. It connects four different artifact types (requirements, system tests, design models, and implementation) with three different trace link types (tested by, realized by, and implemented by).

when using the reference models in a few case studies. Additionally, Mäder et al.<sup>7</sup> give a few guidelines for defining TIMs and provides guidelines for traceability in safety-critical environments, which also include some guidelines on TIM design.<sup>8</sup> Overall, different aspects are still missing, such as alignment of TIMs with different processes and workflows, TIM evolution, and, in general, TIM management in large organizations.

Our long-term goal is to fill this gap. This article presents the first step in this endeavor, i.e., identifying practitioner's challenges when managing

TIMs to synthesize useful solutions and guidelines for these challenges.

### Practitioners' Challenges

The three companies that presented their TIMs in the workshop all have different approaches toward the design of TIMs. (For an overview of methods used, see "Methodology.") In Company 1, the TIM is not formally documented as a model but implicitly defined as part of the definition of "done" and integrated in the continuous integration tooling. Thus, developers are aware of which trace link types to create. The TIM focuses on linking development artifacts in an

agile environment, e.g., test cases, user stories, and code. Company 2 defines an overall information model that shows all artifacts and aspects used in the development environment. The traceability strategy is derived from this information model. Company 3 follows a formal process and defines a company-wide TIM with a centralized approach via a feature model and including traceability in two dimensions: traceability between platform artifacts and between the product instances of the platform artifacts.

While the companies follow different approaches, the challenges can be ascribed to three main drivers:



## METHODOLOGY

We conducted a focus group with five researchers and seven practitioners from three companies from the systems engineering domain. All participants from the companies have experience with traceability. Additionally, participants from two of the three companies have practical experience designing and managing TIMs. During the focus group, one of the researchers presented examples of TIMs and a summary of guidelines on how to define TIMs extracted from the literature. This was followed by presentations from the three companies (Table S1) on their TIM and their TIM management process. Each presentation was followed by a discussion of the specific challenges presented and the relation-

ship to literature. The discussion was open, such that all participants could ask questions about the presented TIM and TIM management process. The researchers took notes during the presentation and discussions.

The notes were later analyzed in a brainstorming session with five researchers. The challenges identified were sorted and grouped into three categories and member-checked with the industry participants of the focus group.

To protect each company's anonymity, we refer to Companies 1–3 when reporting results. These numbers are not in any particular order; specifically, Company 1 does not necessarily correspond to Company A.

**Table S1.** An overview of the three companies that participated in the workshop. To protect each company's anonymity, the text refers to them as Companies 1, 2, and 3 instead.

	Company A	Company B	Company C
Domain	Industrial automation	Telecommunication	Automotive
Number of participants	Three	One	Three
Participants' roles	Two lead systems engineers One system test engineer	One systems engineer	One information architect Two process, method, and tools specialists

- The companies develop complex products using a product line approach.
- The companies are large and distributed, and different teams need to work together on the same end product.
- The companies have long-lasting products, which means traceability needs to be maintained over a long period of time.

The challenges are presented in the following sections and summarized in Figure 1.

## Complex Products

The focus group allowed us to identify challenges that arise due to the complexity of the products developed by the companies.

- *Inherent complexity of TIMs:* A TIM for a large and complex system containing a large number of

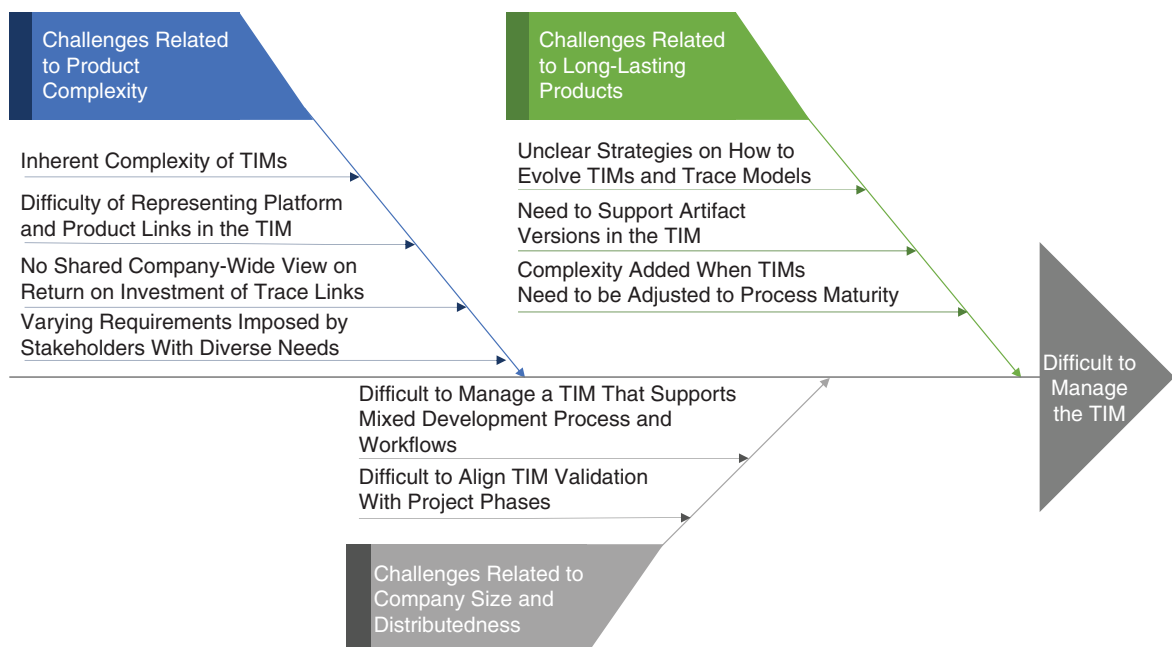
artifact types needs to encompass the entire product and therefore is itself complex with a large number of artifact types and relationship types. Thus, managing such a TIM is difficult.

*Company 2 addresses this challenge by creating an overall information model. It not only defines all artifact types that exist along with their relationship types but also captures additional information, such as tool capabilities and role responsibilities. While one might argue that the information model constitutes a TIM, it contains aspects that are not relevant for traceability of software development artifacts, which are not clearly delineated. For instance, the information model used by Company 2 contains more than 50 artifact types and also defines workflows and responsible roles. This makes it difficult to*

*reason about traceability with such information models.*

*One of the engineers from Company 2 reported that the complexity of the overall information model makes it difficult to define a subset of the information model as a TIM and plan an overall traceability strategy for the company based on this.*

- *Difficulty of representing platform and product links in the TIM:* In a product line environment, traceability needs to support maintenance of the shared assets in the platform as well as the assets for specific products. In such a scenario, companies struggle with defining a TIM in a way that caters to product line activities (e.g., variability tracking and maintenance) and activities related to individual



**FIGURE 1.** An overview of challenges that make design of TIMs difficult in practice.

products (e.g., change impact analysis and project progress monitoring) alike. Ideally, every feature is first implemented on the platform level and only reused at the product level so that trace links from the platform are inherited by specific product instances to achieve full traceability. However, in practice, as reported by Companies 2 and 3, this is rarely the case, and product-specific features exist that make traceability on the product level necessary. It is a challenge to represent platform and product traceability in the TIM to make the traceability as visible and efficient as possible.

- *No shared company-wide view on return on investment of trace links:* Stakeholders defining a TIM need reasons for including certain relationship types in the TIM. A common way to motivate a relationship type is to link it to a stakeholder need by eliciting the traceability needs of the different stakeholders in the company.<sup>7, 8</sup> For smaller systems, it is easier to understand which trace link types are needed, since knowledge of how the different artifact types fit together is common; however, for complex systems, this elicitation leads to many suggested relationship types due to the large number of artifact types. Thus, they need to be filtered by the expected value they create, but it is difficult to predict the concrete value of certain relationship types. Moreover, different link types have different values for different stakeholders. While this is a general traceability challenge, it is also TIM related

since, for complex systems, it is unclear which combination of trace link types provide the most value. All three companies in the workshop struggle with making decisions on which relationship types to include in the TIM. For example, in Company 2, there are more than 50 artifact types, which makes it difficult to decide which trace link types between these artifacts are actually valuable.

- *Varying requirements imposed by stakeholders with diverse needs:* A TIM is typically designed according to the stakeholder needs in the company. For large organizations, a single TIM needs to address the needs of many stakeholders. For instance, different stakeholders may not only require different link types but also require the same link types but on diverse abstraction levels. This results in large and complex TIMs. The representation of such TIMs to stakeholders is difficult, since different stakeholders need to derive different information from the TIM. For example, performing a system-level safety analysis requires trace links on a higher abstraction level than low-level safety analysis of specific functions. In practice, it is challenging to design a TIM that is flexible enough to support these different needs and still provide guidance to developers so that they create useful trace links using such a TIM.

## Large and Distributed Companies

In this section, we describe challenges that are due to the companies being large and distributed but still having

to coordinate and work together to create the same end product.

- *Difficult to define a TIM that supports mixed development processes and workflows:* In large systems engineering companies, different disciplines (e.g., mechanical engineering, electrical engineering, and software engineering) need to be coordinated and integrated. Teams that focus on one of those disciplines tend to follow processes that are aligned with the way the parts of the product are developed. A mechanical engineer, e.g., is more likely to use a waterfallish process, whereas a software engineer might work in an iterative-incremental fashion. Each process requires different kinds of artifacts, e.g., a product requirement document for the mechanical engineer and a backlog item for the software engineer. A TIM that is used in such an organization needs to cater to all processes or workflows and, therefore, integrate all artifacts prescribed by all used processes, independent of discipline, thereby contributing to the TIM's complexity. This challenge was reported by Companies 2 and 3.
- *Difficult to align TIM conformance checks with project phases:* Constraints defined in the TIM, such as cardinality, are used to check if links exist and are syntactically correct. For instance, if the TIM defines a one-to-one cardinality for a link from requirement to test, then this should be enforced to ensure that the link actually exists and the cardinality is adhered to. However, in



practice, the enforcement of such conformance checks needs to be aligned with the current project phase or current phase within the sprint. For instance, the companies reported that enforcing that there is a link between a requirement and a test during requirement elicitation through the TIM is not helpful but is helpful at a later stage when requirements are already gathered and need to be maintained. This challenge becomes more pronounced when mixed development processes are used and there is no clear global definition of phases. Additionally, constraints on the TIM should not limit the agility of the different teams.

### Long-Lasting Products

The products developed by the three companies have to be maintained over a long period of time. Thus, the TIM and the traceability links in place also need to be maintained. This introduces three main challenges with respect to TIM management as organizations, development processes, and systems engineering techniques evolve.

- *Unclear strategies on how to evolve TIMs and trace models:* All of the companies

acknowledged that once the TIM is defined and used, it is hard to change both on a technical and process level since existing links conforming to the current TIM need to be migrated. Additionally, there are no strategies in place to analyze the existing traceability strategy in the companies to decide if the TIM needs to be updated. This leads to TIMs that are defined once and not updated, even though some of the relationships they capture have evolved.

*Currently, TIMs are changed in big bang approaches: When a new platform is created at Company 2, e.g., the TIM is changed as well. The old platform is maintained with the old TIM, and the new platform will be maintained with the new TIM. However, the company now wants to move to a development strategy in which their platform will change less frequently; therefore, the company needs a new strategy on how to continuously evolve the TIM.*

- *Need to support artifact versions in the TIM:* In a typical software and systems engineering product line, various versions of

an artifact coexist and are used in products at the same time. Which version is used in which product needs to be tracked. In the automotive domain, regulations state that the manufacturer needs to trace which versions of the components are deployed in a specific vehicle. Thus, trace links to specific artifact versions need to be established both at the platform and product level (compare to Figure 2).

*Companies 2 and 3 reported that it is a challenge to define a TIM that explicitly supports versioning information for the artifacts connected by trace links.*

*Company 1 reported creating traceability links from requirements and test cases to specific commits but having issues with tracking artifact versions when multiple artifact repositories are used.*

- *Complexity added when TIMs need to be adjusted to process maturity:* The stability of a TIM reflects the stability of the development process. For companies transitioning from one development process to another, TIM management is challenging. Product development continues in the midst of process changes, and traceability still needs to be maintained. In such situations, it is necessary to manage and evolve multiple TIMs in parallel, as different development teams have a variety of schedules for adopting a new development process. For instance, Company 3 is transitioning from single-product development to product-line development. In this transition, three TIMs are

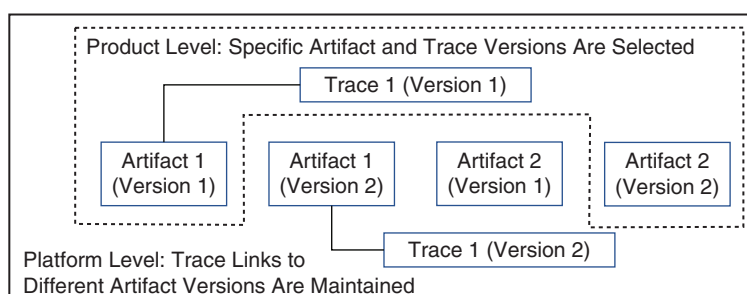


FIGURE 2. The versioning of trace links in a product line environment.

required: a TIM that supports a single product approach, a TIM that supports a product-line approach, and a TIM that supports teams that are “in between,” i.e., teams that have started to transition but are not yet fully transitioned. Such teams end

up creating more traceability links than needed in either of the development paradigms, as they must trace to artifacts in the old and new process. Moreover, as teams migrate from one paradigm to another, trace link migration is difficult due to the

challenge of evolving TIMs and trace models discussed before.

## Refocusing TIM Design Research

The existing guidelines on TIM management, such as those from Ramesh and Jarke<sup>5</sup> and Mader et al.,<sup>7,8</sup> are

## ABOUT THE AUTHORS



**SALOME MARO** is a Ph.D. candidate at Chalmers University of Gothenburg, Gothenburg, 41296, Sweden. Her research is on software and systems traceability tools and processes in industry. Maro received an M.Sc. in software engineering from the University of Gothenburg. Contact her at [salome.maro@cse.gu.se](mailto:salome.maro@cse.gu.se).



**JENNIFER HORKOFF** is an associate professor at Chalmers University of Gothenburg, Gothenburg, 41296, Sweden. Her research focuses on requirements engineering, requirements modeling, quality and value modeling, model analysis, and creativity. Horkoff received a Ph.D. from the University of Toronto. Contact her at [jennifer.horkoff@gu.se](mailto:jennifer.horkoff@gu.se).



**JAN-PHILIPP STEGHÖFER** is an associate professor at Chalmers University of Gothenburg, Gothenburg, 41296, Sweden. His research interests include software and systems traceability, large-scale and distributed agile development, software engineering education, and self-adaptive systems. Steghöfer received a Ph.D. from the University of Augsburg, Germany. Contact him at [jan-philipp.steghofer@gu.se](mailto:jan-philipp.steghofer@gu.se).



**RASHIDAH KASAU LI** is a lecturer at Makerere University, Kampala, Uganda. Her research focuses on requirements, safety-critical systems, and large-scale agile development. Kasauli received an M.Sc. in data communication and software engineering from Makerere University. Contact her at [rashida@chalmers.se](mailto:rashida@chalmers.se).



**ERIC KNAUSS** is an associate professor at Chalmers University of Gothenburg, Gothenburg, 41296, Sweden. His research focuses on managing requirements and related knowledge in large-scale and distributed software projects. Knauss received a Ph.D. from Leibniz Universität Hannover, Germany. He is a member of program and organization committees of top conferences and a reviewer for top journals. Contact him at [eric.knauss@cse.gu.se](mailto:eric.knauss@cse.gu.se).



**REBEKKA WOHLRAB** is with Chalmers University of Technology Gothenburg, Gothenburg, 41296, Sweden, and Systemite AB, Gothenburg, 41664, Sweden. Her research interests include software architecture and knowledge management in large-scale agile development. Wohlrab received her Ph.D. from Chalmers University of Gothenburg, Sweden. Further information about her can be found at <https://rebekkaa.github.io/>. Contact her at [wohlab@chalmers.se](mailto:wohlab@chalmers.se).

## ABOUT THE AUTHORS



**JESPER LYSEMOSE KORSGAARD** is the lead systems engineer and also works with software requirements at Grundfos Holding A/S Bjerringbro, 8850, Denmark. Contact him at [jlysemose@grundfos.com](mailto:jlysemose@grundfos.com).



**NIELS JØRGEN STRØM** is a chief systems engineer at Grundfos Holding A/S Bjerringbro, 8850, Denmark, and has been with the company for 25 years. He has an SE-Zert Level B certification and has a background as an electronics engineer and an International Organization for Standardization assessor. Contact him at [njstroem@grundfos.com](mailto:njstroem@grundfos.com).



**FLORIAN WARTENBERG** is a system test engineer at Grundfos Holding A/S Bjerringbro, 8850, Denmark. His interests focus on requirements, traceability, verification, and validation. Wartenberg received a diploma in computer science from Humboldt University of Berlin. Contact him at [fwartenberg@grundfos.com](mailto:fwartenberg@grundfos.com).



**RUBEN ALEXANDERSSON** is currently a technical specialist at Volvo Cars, Gothenburg, 405 31, Sweden, and has been with the company for 10 years. Alexandersson received a Ph.D. from Chalmers University of Technology and has a background as a teacher in software engineering. Contact him at [ruben.alexandersson@volvocars.com](mailto:ruben.alexandersson@volvocars.com).

a good starting point for practitioners. However, these guidelines do not cover all of the challenges practitioners face when creating TIMs in large systems engineering companies. In particular, there is little support for understanding the implications of the development process aspect on the TIM, especially when mixed development processes are used. Our findings show that this is a problem in practice.

For some of the challenges, e.g., TIMs for product lines, changeability of TIMs, and TIM modularization, research such as that in referenced Anquetil et al.,<sup>9</sup> Paige et al.,<sup>10</sup> and Mäder and Cleland-Huang,<sup>11</sup> respectively, exists. However, this research only investigates the tooling aspect with restricting assumptions, such as that all artifacts persist in certain formats and specific tools are used. This limits

the transferability of such research to practice.

Moreover, some of the guidelines are conflicting in practice. For instance, some literature suggests that the level of abstraction between two artifacts connected by a trace link should match,<sup>7</sup> while organizations that develop safety-critical products need to trace between artifacts on different levels of abstraction to be able to perform various safety analysis tasks required by safety standards. One of the solutions suggested is to trace to different levels of abstraction depending on the use of the trace links and have tool support to represent the links at different levels of abstraction.<sup>5</sup> However, it is not clear how this can be handled in practice in a cost-efficient and effective manner. Additionally, existing TIM design

guidelines are scattered in various publications, making them hard to find, and there is no guidance on how practitioners can operationalize these guidelines.

From our research, we believe that among many possible research directions, the following research is needed in the context of the challenges we discussed:

- Research on managing complex TIMs, such as TIM modularization, use of multiple TIMs in an organization, and adoption of TIMs in a cost-efficient and effective manner
- Creation of guidelines for designing TIMs that support product-line platform-related activities and product-specific activities, as well as how TIMs can support




the derivation of product-specific links from platform links

- How to evolve TIMs and existing trace models, such as using model transformations to transform existing links to match a new TIM and how to evolve TIMs when processes change
- Research on how organizations can design TIMs that support multiple development processes and workflows, e.g., by defining process-agnostic TIMs and providing guidelines on mapping the different artifact types and links to specific development processes.

While some of these topics are not new to the requirements and traceability community, our aim is to explicitly point out these challenges to draw more attention to the practical complexity that exists in companies that deploy traceability. We acknowledge the usefulness of technical and tool-related research on traceability. However, from the perspective of managing TIMs, we believe solving many of the challenges requires a process-oriented research perspective. Therefore, we suggest more empirical studies specifically on TIM management focusing on large organizations developing complex product lines with long-lasting products. Only if research on traceability and TIM management acknowledges these practical challenges will it yield beneficial lessons for practitioners.

**T**his article reports on challenges that practitioners face when managing TIMs. The challenges were identified using a focus group with three companies from the automotive, industrial automation, and telecommunication domain. The challenges are due to three drivers:

complexity of the products developed, the size and distributedness of the companies, and the long-lasting products that need to be maintained. Our study calls for more research on TIM management targeted to companies with such characteristics. 

### Acknowledgments

We thank all company participants for their contributions to this article. This work has been partially supported by Software Center Project 27 on RE for Large-Scale Agile System Development, the Wallenberg Artificial Intelligence, Autonomous Systems, and Software Program, funded by the Knut and Alice Wallenberg Foundation, the Sida/BRIGHT project 317 under the Makerere-Sida bilateral research program 2015–2020, and ITEA 3 Call 6 project 17003 PANORAMA.

### References

1. J. Cleland-Huang, M. Heimdahl, J. H. Hayes, R. Lutz, and P. Maeder, “Trace queries for safety requirements in high assurance systems,” in *Proc. Int. Working Conf. Requirements Engineering: Foundation for Software Quality*. New York: Springer-Verlag, 2012, pp. 179–193. doi: 10.1007/978-3-642-28714-5\_16.
2. M. Kassab, O. Ormandjieva, and M. Daneva, “A traceability metamodel for change management of non-functional requirements,” in *Proc. 2008 6th Int. Conf. Software Engineering Research, Management and Applications*, pp. 245–254. doi: 10.1109/SERA.2008.37.
3. A. Espinoza Limón and J. Garbajosa Sopena, “The need for a unifying traceability scheme,” in *ECMDA Traceability Workshop Proc.*, 2005, pp. 175–184.
4. N. Drivalos, D. S. Kolovos, R. F. Paige, and K. J. Fernandes, “Engineering a DSL for software traceability,” in *Proc. Int. Conf. Software Language Engineering*. New York: Springer-Verlag, 2008, pp. 151–167. doi: 10.1007/978-3-642-00434-6\_10.
5. B. Ramesh and M. Jarke, “Toward reference models for requirements traceability,” *IEEE Trans. Softw. Eng.*, vol. 27, no. 1, pp. 58–93, 2001. doi: 10.1109/32.895989.
6. J. Cleland-Huang, O. C. Gotel, J. Huffman Hayes, P. Mäder, and A. Zisman, “Software traceability: Trends and future directions,” in *Proc. Future Software Engineering*, 2014, pp. 55–69. doi: 10.1145/2593882.2593891.
7. P. Mader, O. Gotel, and I. Philippow, “Getting back to basics: Promoting the use of a traceability information model in practice,” in *Proc. 2009 ICSE Workshop Traceability Emerging Forms Software Engineering*, pp. 21–25. doi: 10.1109/TEFSE.2009.5069578.
8. P. Mäder, P. L. Jones, Y. Zhang, and J. Cleland-Huang, “Strategic traceability for safety-critical projects,” *IEEE Softw.*, vol. 30, no. 3, pp. 58–66, 2013. doi: 10.1109/MS.2013.60.
9. N. Anquetil et al., “A model-driven traceability framework for software product lines,” *Softw. Syst. Model.*, vol. 9, no. 4, pp. 427–451, 2010. doi: 10.1007/s10270-009-0120-9.
10. R. F. Paige, N. Matragkas, and L. M. Rose, “Evolving models in model-driven engineering: State-of-the-art and future challenges,” *J. Syst. Softw.*, vol. 111, pp. 272–280, Jan. 2016. doi: 10.1016/j.jss.2015.08.047.
11. P. Mäder and J. Cleland-Huang, “A visual traceability modeling language,” in *Proc. Int. Conf. Model Driven Engineering Languages and Systems*. New York: Springer-Verlag, 2010, pp. 226–240. doi: 10.1007/978-3-642-16145-2\_16.