Information and Software Technology 68 (2015) 62–81

Contents lists available at ScienceDirect

Information and Software Technology

journal homepage: www.elsevier.com/locate/infsof

An extensive systematic review on the Model-Driven Development of secure systems

Phu H. Nguyen [a,*], Max Kramer [b], Jacques Klein [c], Yves Le Traon [c]

[a] Simula Research Laboratory, Martin Linges vei 25, Fornebu 1364, Norway
[b] Karlsruhe Institute of Technology, Am Fasanengarten 5, Karlsruhe D-76131, Germany
[c] Interdisciplinary Center for Security, Reliability and Trust (SnT), University of Luxembourg, 4 rue Alphonse Weicker, Luxembourg L-2721, Luxembourg

ARTICLE INFO

Article history:
Received 14 April 2015
Revised 31 July 2015
Accepted 24 August 2015
Available online 1 September 2015

Keywords:
Systematic review
Model-Driven Security
MDS
Model-Driven Engineering
MDE
Software security engineering

ABSTRACT

Context: Model-Driven Security (MDS) is as a specialised Model-Driven Engineering research area for supporting the development of secure systems. Over a decade of research on MDS has resulted in a large number of publications.

Objective: To provide a detailed analysis of the state of the art in MDS, a systematic literature review (SLR) is essential.

Method: We conducted an extensive SLR on MDS. Derived from our research questions, we designed a rigorous, extensive search and selection process to identify a set of primary MDS studies that is as complete as possible. Our three-pronged search process consists of automatic searching, manual searching, and snowballing. After discovering and considering more than thousand relevant papers, we identified, strictly selected, and reviewed 108 MDS publications.

Results: The results of our SLR show the overall status of the key artefacts of MDS, and the identified primary MDS studies. For example, regarding security modelling artefact, we found that developing domain-specific languages plays a key role in many MDS approaches. The current limitations in each MDS artefact are pointed out and corresponding potential research directions are suggested. Moreover, we categorise the identified primary MDS studies into 5 significant MDS studies, and other emerging or less common MDS studies. Finally, some trend analyses of MDS research are given.

Conclusion: Our results suggest the need for addressing multiple security concerns more systematically and simultaneously, for tool chains supporting the MDS development cycle, and for more empirical studies on the application of MDS methodologies. To the best of our knowledge, this SLR is the first in the field of Software Engineering that combines a snowballing strategy with database searching. This combination has delivered an extensive literature study on MDS.

1. Introduction

With more and more IT systems being developed and used, approaches for systematically engineering *secure* IT systems are becoming increasingly important. *Model-Driven Security* (MDS) emerged more than a decade ago as a special area of *Model-Driven Engineering* (MDE) for supporting the development of secure systems. MDE has been considered by some researchers as a solution to handle complex and evolving software systems [22]. It leverages *models* and *transformations* as main artefacts at every development stage. MDS specialises MDE by taking security requirements and functional requirements into account at every stage of the development process. By *modelling* and manipulating models, the level of abstraction is higher than code-level that brings several significant benefits, especially regarding security engineering. *First*, security concerns can be considered together with business logic and other quality requirements such as performance from the very beginning, and throughout the MDS development life cycle. *Second*, reasoning about systems at the model level, e.g. with model-based verification and validation methods, makes it possible to check security requirements and other requirements at early design stages. These methods can perform formal verification as well as security testing based on models. Moreover, models that abstract away from target platform details can increase cross-platform interoperability. *Third*, MDS can be more

* Corresponding author.
E-mail address: phu@simula.no (P.H. Nguyen).

http://dx.doi.org/10.1016/j.infsof.2015.08.006
0950-5849/

productive, and supposedly less error-prone than traditional development methods by leveraging automated *model-to-model transformations* (MMTs) and *model-to-text transformations* (MTT s, code generation).

For more than a decade since MDS first appeared, a considerable number of MDS publications have shown a great attention of the research community to this area. The MDS approaches vary greatly in many artefacts such as the security concerns addressed, the modelling techniques used, the model transformations techniques used, the targeted application domains, or the evaluation methods used. To provide a detailed state of the art in MDS, a full systematic literature review (SLR) is needed.

So far, a full SLR on MDS does not exist. Surveys on MDS approaches [15,71,79,121] could provide in-depth analyses of some well-known MDS approaches, but do not summarise the complete research area systematically. [62] could be closer to our work, but has several limitations in terms of scope and methodology. For example, it missed many important primary MDS approaches such as UMLsec [63], and aspect-oriented approaches. In contrast, our SLR is performed in both width and depth of MDS research that reveals an extensive set of primary MDS studies. Furthermore, our review provides a detailed overview on key artefacts of every MDS approach such as used modelling techniques, considered security concerns, employment of model transformations, verification or validation methods, and targeted application domains. Finally, we present trend analyses for MDS publications, and for the addressed security concerns and other key artefacts.

This paper is an extended and improved version of [101]. In the previous version, we reported the results of a SLR based on 80 MDS papers found from an automatic search and a rigorous selection process. In this extended version, we improved our set of primary MDS papers by conducting two more search strategies: manual search and snowballing. On the resulting set of 108 finally selected MDS papers, we performed more detailed analyses for key artefacts, primary MDS studies, and trend analyses for a period of more than a decade.

The main contributions of this paper are: 1) detailed and condensed results on key MDS artefacts of all identified primary MDS publications; 2) a diagnosis of limitations of current MDS approaches with suggestions for potential MDS research directions; 3) a classification of significant and emerging/less common MDS approaches; and 4) trend analyses.

The remainder of this paper is structured as follows. Section 2 provides some main background concepts and definitions that are used in this paper. The objective of this SLR, its research questions, search strategy, and selection process are described in Section 3. In Section 4, we present our evaluation criteria and data extraction strategy. Section 5 shows the main results of our review. Threats to validity are discussed in Section 6. In Section 7, we position this work regarding related work. Section 8 concludes the paper by summarising the results, highlighting open issues, and giving some thoughts on future work.

2. Background concepts and definitions

2.1. Systematic literature review and snowballing

SLR is a means for thoroughly answering a particular research question, or examining a particular research topic area, or phenomenon of interest, by systematically identifying, evaluating, and interpreting all available relevant research [77]. Well-known guidelines for conducting SLR sin software engineering were provided by Kitchenham [77] and Biolchini et al. [23]. All individual studies that are identified as relevant research contributing to a SLR are called *primary* studies [77]. In this paper, based on the numbers of publications and citations of *primary* MDS studies, we further classify them into *significant* MDS studies, and *less common* or *emerging* MDS studies.

In a SLR, it is crucial to transparently and correctly identify as many relevant research papers in the focus of the review as possible. The search strategy is key to the identification of primary studies and ultimately to the actual outcome of the review [128]. The guidelines by Kitchenham [77] for SLR sin software engineering suggest to start with a database search that is based on a search string and also called *automatic search* in this paper. They also recommend complementary searches, e.g. a *manual search* on journals and conferences proceedings, references lists, and publications lists of researchers in the field.

Both automatic search and manual search have limitations [128]: the former depends on the selection of databases, on database interfaces and their limitations, on the construction of search strings, and on the identification of synonyms. The latter depends on the selection of research outlets, e.g. journals or conferences, and cannot be exhaustive. Therefore, Wohlin and Prikladnicki [128] proposed the snowballing search strategy as a first step to systematic literature studies. The key actions of the snowballing search strategy are: 1) identify a starting set of primary papers; 2) identify further primary papers using the reference lists of each primary paper (backward snowballing); 3) identify further primary papers that cite the primary papers (forward snowballing); 4) repeat Steps 2 and 3 until no new primary papers are found. We are convinced, that the snowballing search strategy complements the automatic and manual search strategies of [77]. In our SLR we defined and performed a snowballing search strategy that builds on the set of primary papers found in automatic and manual searches. Details of our search strategy are presented in Section 3.

2.2. A definition of MDS

Numerous security engineering techniques exist which support the development of secure systems. There are also many MDE techniques for the development and maintenance of software systems in general. Our focus, however, is only on MDE approaches that are specifically customised for supporting the development of secure systems. As we already mentioned, MDS can be considered a subset of MDE. We will now clarify the relations between MDE, Model-Based Engineering (MBE), Model-Driven Development (MDD), security engineering, and MDS, which are important for our inclusion and exclusion criteria (Section 3.3). Regarding MBE, MDE, and MDD, we agree with the point of view presented by [31, p. 9]. Specifically, MBE can be used for development processes in which models may not necessarily be the central artefacts for development. For example, if models are only used for documentation purposes and not in automated transformations. MDE can be seen as a subset of MBE in which models have to be the key artefacts throughout the development, i.e. models "drive" the process in every step. In other words, MDE is truly model-driven in every task of a complete software engineering process. This means that all development, evolution, and migration tasks have to be influenced by explicit models. Regarding MDE, model-to-model transformations (MMTs) or model-to-text transformations (MTTs) could be used by an MDE approach not only in development phase, but also in evolution or migration phases. MDD can be considered a subset of MDE that only denotes development activities with models as the primary artefact. Normally, MMTs and MTT s are used in MDD to obtain other models or to generate code in development activities. The core part of a MDD process includes modelling/designing phase which could lead to code generation phase. Other activities such as requirement engineering, testing might be also included. Regarding MDS, security-oriented models is a key artefacts. MMTs and MTT s could be used to manipulate security-oriented models in the MDS activities. Thus, MDS refers to all research approaches that focus on a