Not Ready for Prime Time:
A Survey on Security in Model Driven Development

Jostein Jensen, Norwegian University of Science and Technology, Norway

Martin Gilje Jaatun, SINTEF, Norway

ABSTRACT

Model Driven Development (MDD) is by many considered a promising approach for software development. This article reports the results of a systematic survey to identify the state-of-the-art within the topic of security in model driven development, with a special focus on finding empirical studies. The authors provide an introduction to the major secure MDD initiatives, but the survey shows that there is a lack of empirical work on the topic. The authors conclude that better standardization initiatives and more empirical research in the field is necessary before it can be considered mature.

Keywords: *Model Driven, Model Driven Architecture (MDA), Model Driven Development (MDD), Security, Software Development*

1. INTRODUCTION

Model Driven Development (MDD) has been considered a promising approach to software development since its introduction about a decade ago. The Object Management Group (OMG, 2010) is the most prominent standardization body within the MDD domain, and has developed a framework for model driven development called Model Driven Architecture (MDA). MDA is a framework for developing applications and writing specifications, where improved portability, platform independence and cross-platform interoperability are among keywords used by OMG to describe the benefits of using this framework.

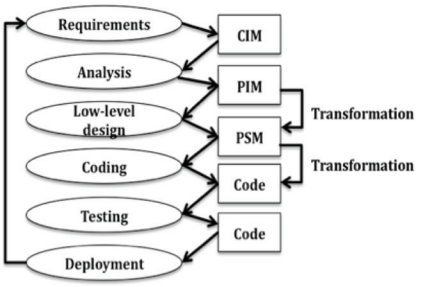
Kleppe et al. (2003) present the MDA development lifecycle. The basis for development is platform independent models (PIM), which specify functionality and behavior. These models are abstracted away from the technology that will be used to realize the system. PIMs can then be transformed into platform specific models (PSM), adding technology specific details to the PIM. PSM again can then be transformed into code. Kleppe and colleagues also mention a third model type used during the requirements and analysis phase of development, called computational independent model (CIM).

Figure 1 shows the MDA software development lifecycle as it is depicted by Kleppe et al. (2003). The ovals to the left represent generic software development phases, while the squares to the right represent artifacts produced in an

DOI: 10.4018/jsse.2011100104

Copyright © 2011, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

Figure 1. MDA Software development lifecycle



MDA context. Artifacts developed during the requirements phase and used for analysis are often referred to as Computational Independent Models (CIM). Platform independent models (PIM) are abstract representations of the system to be built, and independent of any implementation technology. PIMs are transformed, preferably automatically using tool support, to Platform Specific Models. These are specific to the technology that will be used to realize future systems. Continuing the MDA lifecycle, PSMs are transformed into code. Since PSMs are close to the technology, this transformation is by some considered to be straightforward (Kleppe et al., 2003).

Note that real life seldom has a perfect match for theoretical frameworks such as the MDA lifecycle presented in Figure 1. Thus, in concrete examples one will not always find that all the models such as CIM, PIM and PSM are actually used in practice, and in such cases one must modify the map to fit the terrain.

PIMs form the basis for low-level system designs and as such constitute an important part of a system's documentation (while still providing important abstractions). The layering between platform independent models, platform specific models and code are the key to solve problems related to portability, platform independence and interoperability. Developers are mainly supposed to work with the platform independent models, and since these are platform and technology neutral it should be a relatively simple task to transform them into different platforms and technology solutions.

In traditional software development, security aspects are often considered late in the development lifecycle, if they are considered at all (Wyk & McGraw, 2005). However, the cost of eliminating security flaws increases by magnitudes the later they are discovered and fixed (Boehm & Basili, 2001). A good recommendation has therefore been to include security aspects from the very start of software projects (Tøndel, Jaatun, & Meland, 2008). The Microsoft Security Development Lifecycle (Howard & Lipner, 2006) and McGraw's touchpoints (McGraw, 2006) illustrate how security activities can be included in every phase of a software project.

With its focus on high-quality design in early development phases through detailed PIM modeling, MDD/MDA should be a well suited development framework to include security aspects in design models from the very start of a project. Consistent and sound security solutions throughout the entire application could be the result.

The remainder of this article is organized as follows: In Section 2 we present our research questions, followed by a description of our research method in Section 3. We present our results in Section 4, and discuss our findings in Section 5. Section 6 concludes the article.

2. RESEARCH QUESTIONS

This article reports results related to a systematic survey that was carried out in order to learn how scientific communities deal with security

Copyright © 2011, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

in model driven development. The study aims to answer the following research questions:

RQ1 What are the major scientific initiatives describing automatic code generation from design models within the context of security in MDD?
RQ2 What empirical studies exist on the topic "security within MDD/MDA"?
RQ3 What are the strengths of the evidence showing that security aspects successfully can be modeled as an inherent property and transformed to more secure code?

3. METHOD

A systematic literature review approach (Kitchenham, 2004) is used as research method leading to the results presented in this article. This method requires rigor with respect to planning, conducting, and reporting the review. The aim of this systematic survey was to identify scientific literature that could provide answers to our research questions listed in the previous section.

3.1. Identification of Research

The starting point for the survey is a research protocol where the research questions and the search strategy are defined. To support the paper selection process, the protocol also specifies inclusion and exclusion criteria. A rigorous and comprehensive search is key to identify all the relevant scientific literature. Both sources for scientific literature and search phrases were specified prior to the search. We used four online databases for scientific literature to search for studies:

- IEEE Xplore (http://ieeexplore.ieee.org/)
- ACM Digital Library (http://portal.acm.org/dl.cfm)
- ISI Web of Knowledge (http://apps.isiknowledge.com)
- Compendex (http://www.engineeringvillage2.org/)

According to experiences made by Dybå et al. (2007), this should be sufficient to find relevant literature within the information systems field. The use of other databases will lead to duplicate findings, and as such, lead to extra work. For each of these databases we used the following search phrase and keywords:

1. "Model driven development"
2. "Model driven architecture"
3. MDD
4. MDA
5. Security

These were combined as follows: (1 OR 2 OR 3 OR 4) AND 5.

The searches were performed March 12, 2010, meaning that scientific literature indexed up until then are included within this study. The search resulted in a total of 2844 titles that needed to be evaluated based on title, abstract and content. We performed a follow-up search in June 2011, which yielded an initial result of 27 titles (these are treated separately).

3.2. Selection of Primary Studies

All references and abstracts were imported to the reference tool EndNote. The next step was to exclude papers based on titles. All titles that clearly did not treat the wanted topics were filtered out. After this process a total of 366 studies remained. The following task was to read through the abstracts of these papers and evaluate whether they were relevant or not. For both these steps the following exclusion criteria were used:

- Exclude everything that is clearly not related to model driven software development.

Our research interest is on use of models to generate code. Some studies e.g., present research where MDD principles are used to generate firewall rules. Such studies were excluded.

Copyright © 2011, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

- Exclude everything that clearly not concerns both model driven development and security research.

122 papers remained after reading the abstracts, and these papers were all read to make a final evaluation whether they should be part of our primary studies or not. This evaluation resulted in 56 remaining papers. A last exclusion criterion was used for the purpose of this article in order to answer RQ1:

- Exclude studies by authors and research groups who have published 3 or fewer papers on the topic.

There is a chance that this exclusion criterion can give a somewhat inaccurate view of the current state, as some important initiatives conceivably could be treated and enhanced by a large number of research groups, but where each individual group has not published more than 3 papers. For the purpose of this article, it is however considered sufficient to give a rough idea about the current state. With this last exclusion criterion the number of papers to include as primary studies in this report was limited to 30.

3.3. Quality Assessment, Classification and Synthesis

RQ2 and RQ3 can only be answered with a scientific validity if empirical studies following a rigorous research protocol on the topic are found. However, within the topic of model driven development and security, this study shows that no empirical studies seem to exist. Within this article we therefore give a short introduction to the included papers considered for answering RQ1. Studies are grouped based on the originating research groups. A qualitative reflection about how MDD and security is covered in existing research works is given at the end of this article.

4. RESULTS

Our survey identified 5 research approaches which will be described in the following.

4.1. Model Driven Security

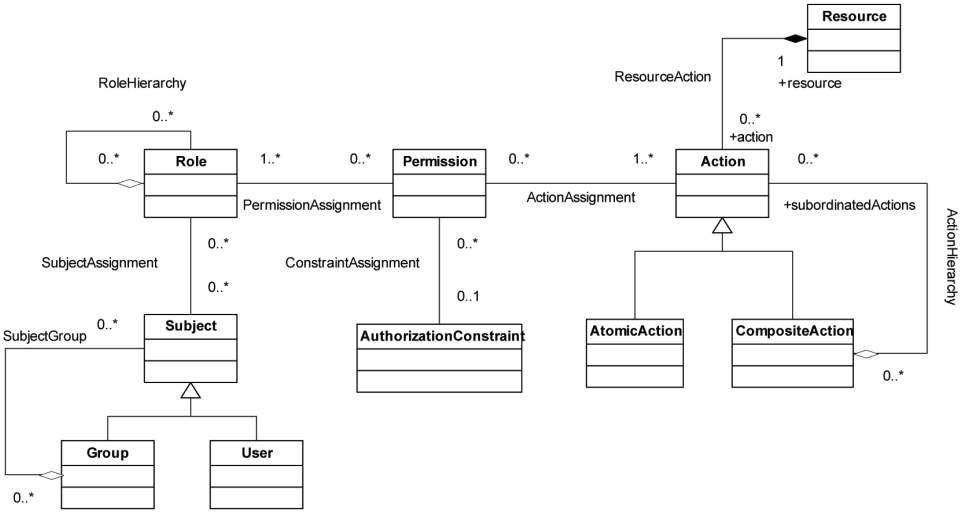
One of the earliest initiatives for including security in model driven architecture came from Basin et al. (2003). Their solution, called Model Driven Security (MDS), is a specialization of the MDA approach. Security models are integrated with what Basin et al. call UML process models, and the combined models are transformed into executable systems with integrated security infrastructures. The focus of their work is to include access control constraints based on role based access control (RBAC) in design models. They describe a security metamodel for expressing RBAC properties in UML, and this UML extension is called SecureUML (see Figure 2, adapted from Basin et al., 2006). Basin et al. (2006) give a more detailed description of the Model Driven Security approach, while Clavel et al. (2008) build on this work to gain practical experience with the approach. See also Section 5.1 for more discussion on MDS.

4.2. SECTET

Alam et al. (2004) describe an approach to specify role-based access control policies for web services using the Object Constraint Language (OCL). OCL was initially a language extension of UML and is used to ensure a platform independent specification of access control policies. This work is used and extended by Breu et al. (2005) who show how security can be built into web service-based systems supporting inter-organizational workflows. To model inter-organizational workflows they specify three model levels: global workflow model, local workflow model and interface model. The global workflow models show an abstract view of interactions between autonomous organizations, the local workflow models show

Copyright © 2011, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

Figure 2. SecureUML metamodel



intra-organizational workflows within each organization, and the interface models present the services offered by each component in the system. OCL is used together with the interface models to describe access control constraints for operations/services provided by a web service. The same team builds on these concepts in a later publication (Hafner, Breu, Breu, & Nowak, 2005) where the focus is to integrate security into the global workflow model. They use OCL-like expressions to assign security qualities such as confidentiality and integrity to data sent between actors.

The research team behind the above mentioned reports (Alam et al., 2004; Breu et al., 2005; Hafner et al., 2005) has built on these results and come up with a model driven security framework called SECTET. Three software engineering paradigms are combined in this framework (Hafner & Breu, 2009): Model Driven Architecture as methodical concept, Service Oriented Architecture as architectural paradigm, and web services as technical standard. The three model levels described above is kept, and the OCL security policy definitions are refined into an OCL-based language they call SECTET-PL. Alam et al. (2006) present the SECTET framework with a focus on integrating access control policies in the interface models. They give a detailed description on how they specify dynamic access control constraints using SECTET-PL, and how these policy rules are combined with UML models at the interface level. In Alam, Hafner, Breu, and Unterthiner (2007) SECTET-PL is used to describe how delegation rights in service-oriented architectures can be implemented, and in Alam, Breu, and Hafner (2007) and Alam, Seifert, and Xinwen (2007) SECTET is presented in a trust management perspective.

While the early reports (Alam et al., 2004; Breu et al., 2005; Hafner et al., 2005) only were at the idea phase (Alam et al., 2006) describes the whole tool chain to carry out model-to-model transformation and model-to-code transformation. They define UML meta-models for their concepts to formalize the modeling process sufficiently to allow tool-supported transformations, and Hafner et al. (2006) focus on using the OMG transformation specification Meta Object Facility Query/View/Transformation (MOF-QVT: http://www.omg.org/spec/QVT/1.1/Beta2/) to formalize transformation rules.

Copyright © 2011, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

Fernandez-Medina et al. (2009) describe the SECTET-framework to be one of the most complete frameworks to integrate security engineering with Model Driven Architecture.

4.3. Secure Development of Data Warehouses

Data warehouses (DW) are repositories where enterprises electronically can store data from their various business systems (http://searchsqlserver.techtarget.com/definition/data-warehouse). This is done to facilitate reporting and analysis of the data. Often data is "… extracted from multiple heterogeneous, autonomous, and distributed sources of information" (Soler, Trujillo, Fernandez-Medina, & Piattini, 2007b). Single data elements in the repository can be sensitive, but also the total amount of business information collected soon becomes business sensitive. Soler et al. (2007b) therefore argue that security engineering must be included from the earliest phases of development of such systems. Soler and his colleagues (Soler et al., 2007b; Soler, Trujillo, Fernandez-Medina, & Piattini, 2007c; Soler, TruJillo, Fernandez-Medina, & Piattini, 2007a) argue that MDA is a well suited development framework to create DW solutions, but with the disadvantage that the MDA framework does not include mechanisms to sufficiently express security requirements (it may be argued that such mechanisms were never intended to be a part of MDA), and as such perform a transformation from PIM to PSM. In their work based on the UML modeling language they show how they use UML profiles and model a security enriched PIM meta-model for the DW domain. In their framework, they also provide a set of QVT transformation rules so that PIMs can be transformed and mapped to concepts in a security enriched PSM meta-model that they also have defined. In addition to the security concepts defined in the two meta-models, dynamic security rules, such as audit and authorization rules can be added to the model using the OCL language. While Soler et al. focused on PIM to PSM transformations, Blanco et al. (Blanco, de Guzman, Fernandez-Medina, Trujillo, & Piattini, 2008; Blanco, Fernandez-Medina, Trujillo, & Piattini, 2008) build on this work and demonstrate with a prototype that it is feasible to go all the way in the MDA lifecycle, from secure PIMs to secure PSM to code with security properties, in order to build secure data warehouses. Soler et al. (2009) supplement this work.

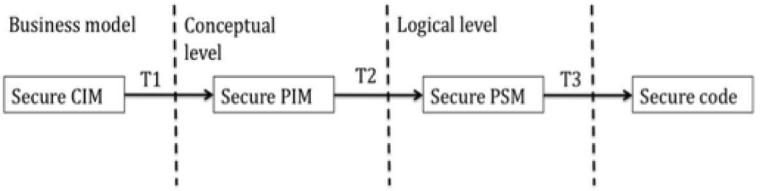
The framework for development of secure data warehouses is further extended in Soler et al. (2008) and Trujillo, Soler, Fernández-Medina, and Piattini (2009b). In these works the authors build on the i* modeling language, which is designed to support modeling of business requirements. i* concepts are converted to a UML profile to fit the DW MDA approach, and some extensions are made to the original i* concepts to be able to sufficiently express security requirements in the DW domain. This new i* UML-profile supports elicitation of requirements at the business level, and is considered as being a CIM. Guidelines for transforming the business security requirements models to PIM are given to align the approach with MDA.

Blanco et al. (2009) present an approach for modernizing existing DWs by means of the above mentioned techniques for secure DW development. By going backwards in a reverse engineering style, they claim that code for existing DWs, presumably with insufficient security, can be analyzed and converted to a PSM. This PSM is again transformed into a PIM, and finally a CIM. Now, the CIM can be analyzed from a business perspective. Security requirements can be added, and then the new secure DW approach can be followed to get a more secure DW with the same functionality as it had before modernization.

To bring the secure DW MDA approach closer to completion, Trujillo et al. (2009a) define an engineering process to support the framework shown in Figure 3 (adapted from Trujillo et al., 2009a). This paper defines the process that starts with i* -based CIM models, which are transformed into secure PIMs, PSMs and code through transformation T1 to T3. It shows that security can be included from the very beginning of a project by using an MDA approach.

Copyright © 2011, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

Figure 3. Framework for designing secure data warehouses



4.4. Security in Business Process Models

Rodriguez et al. (2006a, 2006b) present initial ideas on how UML 2.0 activity diagrams, which are used to model business processes, can be enriched to include security properties. The authors claim that the advantage of including security in the business process modeling stage is that this important aspect then can be included from the very beginning of a software development project, and that a business analyst's considerations about security can be captured. They define a UML profile consistent with OMG MOF, similar to the ideas of Hafner et al. (2006). A graphical notation to represent security requirements is added to the activity diagram notation. In Rodriguez, Fernandez-Medina, and Piattini (2007) the same authors suggest how the business process models, which they consider to be CIMs, can be transformed into use case models, which they consider to be PIMs. The transformation process is based on OMGs QVT specification, checklists and refinement rules. The feasibility of the approach is demonstrated through a prototype tool (Rodriguez, Fernandez-Medina, & Piattini, 2008). Use case models are often the starting point in software development projects where they are used to capture functional requirements. With this work, functional security requirements can be visually illustrated from the start in these models.

4.5. Secure Smart Card Application Development

Moebius et al. (2009a, 2009b) and Moebius, Stenzel, and Reif (2009) use a model driven approach, which they call secureMDD, to develop security critical applications for smart cards. Their illustrating case is the development of an application that can be used for payment. From the PIMs they design, a transformation to three new model types is made: to card PSM, to terminal PSM and to a formal PSM. The two first model types define the functionality on, and interaction between the payment card and the terminal in which the card is used. The latter is a formal security specification of their models that can be analyzed to determine the correctness with respect to security of their models.

Moebius and her colleagues emphasize the importance of both modeling static and dynamic aspects of the application. UML is the preferred modeling language in their approach.

The secureMDD approach is introduced in Moebius, Stenzel et al. (2009b), and the approach to go from PIM to PSM to code is specified in more detail in Moebius, Stenzel et al. (2009a). Class diagrams are used to model an application's static view, while sequence and activity diagrams are used for modeling of dynamic aspects. The transformation from PIMs to formal specifications is shown in Moebius, Stenzel, and Reif (2009).

4.6. Recent Contributions

We repeated the search procedure in June 2011, in order to determine if studies published after the acceptance of our SecSE paper (Jensen & Jaatun, 2011) were likely to affect our results. The initial search yielded an additional 27 papers, but on closer examination, none of the papers provided significantly new information. However, we find it prudent to mention two survey articles published in 2011. The first

Copyright © 2011, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

is published by the group responsible for the Model Driven Security concept (Basin, Clavel, & Egea, 2011), and is primarily a stock-taking of their own work over the last decade.

The second survey is an independent contribution (Kasal, Heurix, & Neubauer, 2011) which in one way is complementary to our own, in that none of the approaches they discuss satisfy the "executability" criterion (see Table 1 in their paper). In our opinion, their survey is less focused than our own, since they cover both UMLsec (Jürjens, 2005), aspect-oriented approaches and formal security protocol analysis tools. It is thus not surprising that they conclude that none of the approaches have general applicability, and different approaches may be most suitable depending on the case at hand.

5. DISCUSSION

The premise of MDD/MDA is that developers will use model-based tools to develop general software. This premise was challenged during the audience discussion after the presentation of our SecSE paper (Jensen & Jaatun, 2011), and opinions were voiced to the effect that MDD will only be used for software deemed to be particularly critical with respect to safety or security. Several members of the audience expressed doubts to whether MDD/MDA will ever find its way to the mainstream developer community.

5.1. What Can We Learn from the Literature?

The existing papers on the topic can be categorized as lessons learned or experience reports, where approaches are demonstrated primarily by implementing prototypes. They provide little evidence to prove that the final code is more secure or better than what it would have been if another development approach had been used. The contribution that comes closest to being an empirical study is the paper written by Clavel et al. (2008). They provide an experience report where the Model Driven Security (MDS) approach defined by Basin et al. (Section 4.1) has been tested in an industrial setting. Their feedback on the approach is quite optimistic, and with respect to MDS their major findings are

- "The security design models integrate security models with system design models, remaining at the same time technology independent, reusable, and evolvable."
- "The security design models are understandable by those familiar with the UML-notation"
- The security-enhanced models were "expressive enough to model the access control policy defined in the original requirements document" provided by their clients.

This seems promising, but there are still several challenges that should be addressed in the coming years. Some of the promises of MDD/MDA are that the approach will ensure portability, platform independence and cross-platform interoperability. However, the studies included in this article all explain different approaches for including security into the modeling languages and the processes they use. Since it is recognized that security modeling is not part of any standardization initiatives for MDD, e.g., MDA, researchers define their own extensions to existing modeling languages to model the security aspects they need for their projects. An example of this is the use of OCL, which is the standardized UML constraint language used as starting point for specifying dynamic security aspects in the two most complete MDA frameworks: SECTET and secure DW. Both research teams found limitations with respect to modeling security constraints in the OCL language. Consequently, they started adapting it. In the SECTET framework, the SECTET-PL was the resulting constraint language used, and in the DW design they extended a DW UML profile in order to better integrate concepts from the OCL expressions into their models. In general, standardization initiatives exist with the purpose of encouraging the development of interoperable systems, so when standards are adapted and extended in different ways by different research teams it can be questioned

Copyright © 2011, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

whether final systems really will be interoperable and portable and so on.

McDermott (2005) argues that one topic not sufficiently covered within security modeling, is related to modeling of security protocols. Moebius and her colleagues treat this in their approach for secure smart card application development. However, they do not follow a standardized MDD approach such as the MDA framework. At the same time it can be questioned whether the descriptions of their approach is sufficient to reconstruct their transformations from PIMs expressing protocol information to PSMs and then code. Thus, McDermott's point still seems to be valid.

A key ingredient in MDD is the transformation rules guiding conversion from PIM to PSM to code. Based on the papers included in this study, the transformation rule development seems like a complex task, which requires a lot of expertise both with respect to the used development approach and technology platforms. This raises questions whether the team of security experts responsible for analyzing security needs and requirements, also need to be experts on the modeling approach. If a transformation rule is flawed in a sense that it does not correctly transform a security requirement/model to code, then the whole system's security can be compromised. Security experts should therefore also be able to evaluate the quality of transformation rules in all parts of the transformation chain to successfully benefit from the promises of security in MDD. Unfortunately, the situation seems to be that development teams and security teams often are separated, and that the real security experts usually do not themselves develop software (Wyk & McGraw, 2005). This situation must be changed if high-quality secure code is going to be produced in an MDD context with automated code generation.

There is one important topic related to security that has not been discussed in the papers identified in this study; the possibility to model input validation constraints. Data sent to interfaces should be validated before they are accepted. Both the length and type of data must be checked in order to avoid security vulnerabilities related to injection attacks. To date, these types of vulnerabilities are the most prevalent security flaws in existing web applications (OWASP, 2011). It should be possible to include modeling of input validation constraints in order to eliminate injection attack threats from the start of software development, similar to modeling of access control constraints.

5.2. Excluded Studies

There have been significant initiatives on topics related to this study that have been excluded due to RQ1 and the exclusion criteria used for the purpose of selecting primary studies. A notable example is UMLsec, an extension of UML supporting secure systems development (Jürjens, 2005). Security requirements such as confidentiality, integrity and authenticity can be modeled in UML diagrams through the extension mechanisms stereotypes and tags. Modeling with UMLsec and analysis of industrial systems using this approach is even tested in industrial projects (Best, Jurjens, & Nuseibeh, 2007; Jürjens, Schreck, & Bartmann, 2008; Lloyd & Jürjens, 2009). However, even though UMLsec is an important contribution to security engineering research in general and in the core of security in model driven development, papers on this topic were excluded due to our focus on automatic code generation.

Another topic not dealt with in this study is aspect oriented modeling. In aspect oriented modeling crosscutting concerns for an application, or aspects, are treated separately. Each aspect is then modeled and, by tool support, woven together into the final product. Examples of what an aspect might be include security, mobility and availability. Aspect oriented modeling papers were excluded since security was not treated specifically, but as one of several aspects. Still, we recognize that this approach may be worth looking into in future studies.

In the past there have been attempts to identify empirical research on the wide topic of model driven development. The systematic survey performed by Haug (2007) returned a total of 21 papers, but this was only 2.2% of the

Copyright © 2011, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

studies from the initial search; none with special focus on security. There were, however, limitations in this study with respect to sources used to find relevant literature; only selected journals and conference proceedings were searched. One of the key objectives of the review presented in this article was to identify empirical studies on the topic of security in MDD, which is a narrow field compared to what Haug presented. A search strategy with a wider scope with respect to publication databases was used in hope of finding relevant literature despite the findings by Haug. However, the observations made in our study (including the studies that are not presented in this article) indicate that such empirical studies do not exist for the topic of security in Model Driven Development.

5.3. Further Work

From the discussion above, the following paths for future research are identified:

- Empirical research should be performed to determine whether security successfully can be included properly in MDD/MDA to build more secure systems.
- Modeling of security should be included as a standardization activity in the MDD frameworks, such as MDA.
- More research should be performed related to how security protocols can be modeled and transformed to final systems.
- Research should be performed to find an approach for modeling of input validation constraints.

Additionally, a follow up of what is presented within this article seems natural. Here we have presented an introduction to the major initiatives within the field of security in MDD. Future work must cover a deeper analysis, which includes evaluating the maturity of the presented approaches, to see if they are ready to be applied within an industrial setting. It is also worth studying the main differences and commonalities of each approach to determine to what extent their elements can be combined or reconciled. Finally, it is worth looking into a refinement of the research protocol, maybe widen the scope of the research questions and exclusion criteria, so that initiatives such as UMLsec and Aspect Oriented Modeling will be covered.

A more fundamental challenge, however, resides in the area of measuring code security, i.e., comparing two pieces of code to determine which is most "secure". Current approaches are limited to counting the accumulated number of discovered bugs/flaws in a software product (CVE, 2011), or (reverse) modeling a given implementation and comparing it to an "ideal" model (Best et al., 2007; Jürjens et al., 2008; Lloyd & Jürjens, 2009) the latter approach assumes that the "ideal" model always will produce more secure code, but unless you can measure the security property, there is no way to know for sure. It is not clear whether this problem is solvable, and we are not aware that anyone is currently working on it. The authors of UMLsec state that "Automated theorem provers and model checkers automatically establish whether the security requirements hold" (Jürjens et al., 2008), but this is no panacea if the security requirements themselves are flawed (or missing).

In order to minimize potential bias in this study, the inclusion and exclusion criteria in the research protocol have been followed stringently. Yet, it is recognized that this is not the optimal situation. The consequences of this threat to validity would, however, have been more severe in a study where empirical evidence would have been subject to quantitative meta-analysis. This article reports the state-of-the art merely, and provides a qualitative reflection on this.

6. CONCLUSION

In this article we have presented state-of-the art within security research in model driven development and identified the most comprehensive works. The study shows that there is a need for more empirical studies on the topic,

Copyright © 2011, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

and we believe that standardization is key to achieve the objectives of MDD/MDA, which are increased portability and interoperability.

REFERENCES

Alam, M., Breu, R., & Breu, M. (2004). Model driven security for web services (MDS4WS). In *Proceedings of the 8th International Multitopic Conference* (pp. 498-505).

Alam, M., Breu, R., & Hafner, M. (2007). Model-driven security engineering for trust management in SECTET. *Journal of Software*, *2*(1). doi:10.4304/jsw.2.1.47-59

Alam, M., Hafner, M., & Breu, R. (2006). Constraint based role based access control (CRBAC) for restricted administrative delegation constraints in the SECTET. In *Proceedings of the International Conference on Privacy, Security, and Trust: Bridge the Gap between PST Technologies and Business Services*, Markham, ON, Canada.

Alam, M., Hafner, M., Breu, R., & Unterthiner, S. (2007). A framework for modelling restricted delegation of rights in the SECTET. *Computer Systems Science and Engineering*, *22*, 289–305.

Alam, M., Seifert, J. P., & Xinwen, Z. (2007). A model-driven framework for trusted computing based systems. In *Proceedings of the 11th IEEE International Enterprise Distributed Object Computing Conference* (pp. 75-75).

Basin, D., Clavel, M., & Egea, M. (2011). A decade of model-driven security. In *Proceedings of the 16th ACM Symposium on Access Control Models and Technologies.*

Basin, D., Doser, J., & Lodderstedt, T. (2003, June 2-3). Model driven security for process-oriented systems. In *Proceedings of the 8th ACM Symposium on Access Control Models and Technologies*, Villa Gallia, Como, Italy.

Basin, D., Doser, J., & Lodderstedt, T. (2006). Model driven security: From UML models to access control infrastructures. *ACM Transactions on Software Engineering and Methodology*, *15*(1), 39–91. doi:10.1145/1125808.1125810

Best, B., Jurjens, J., & Nuseibeh, B. (2007). Model-based security engineering of distributed information systems using UMLsec. In *Proceedings of the 29th International Conference on Software Engineerin*g.

Blanco, C., de Guzman, I. G. R., Fernandez-Medina, E., Trujillo, J., & Piattini, M. (2008). Automatic generation of secure multidimensional code for data warehouses: An MDA approach. In R. Meersman & Z. Tari (Eds.), *Proceedings of the International Conference of On the Move to Meaningful Internet Systems* (LNCS 5332, pp. 1052-1068).

Blanco, C., Fernandez-Medina, E., Trujillo, J., & Piattini, M. (2008, March 4-7). Implementing multidimensional security into OLAP tools. In *Proceedings of the 3rd International Conference on Availability, Security, and Reliability*, Barcelona, Spain.

Blanco, C., Pérez-Castillo, R., Hernández, A., Fernández-Medina, E., & Trujillo, J. (2009). Towards a modernization process for secure data warehouses. In T. B. Pedersen, M. K. Mohania, & A. M. Tjoa (Eds.), *Proceedings of the 11th International Conference on Data Warehousing and Knowledge Discovery*, Linz, Austria (LNCS 5691, pp. 24-35).

Boehm, B., & Basili, V. R. (2001). Software defect reduction top 10 list. *IEEE Computer*, *34*, 135–137.

Breu, R., Hafner, M., Weber, B., & Novak, A. (2005, March 2-4). Model driven security for inter-organizational workflows in e-government. In M. Böhlen, J. Gamper, W. Polasek, & M. A. Wimmer (Eds.), *Proceedings of the International Conference on E-Government: Towards Electronic Democracy*, Bolzano, Italy (LNCS 3416, pp. 122-133).

Clavel, M., Silva, V., Braga, C., & Egea, M. (2008). Model-driven security in practice: An industrial experience. In *Proceedings of the 4th European Conference on Model Driven Architecture: Foundations and Applications* (pp. 326-337).

CVE. (2011). *Common vulnerabilities and exposures (CVE).* Retrieved from http://cve.mitre.org/

Dybå, T., Dingsøyr, T., & Hanssen, G. K. (2007). Applying systematic reviews to diverse study types: An experience report. In *Proceedings of the 1st International Symposium on Empirical Software Engineering and Measurement* (pp. 225-234).

Fernandez-Medina, E., Jurjens, J., Trujillo, J., & Jajodia, S. (2009). Model-driven development for secure information systems. *Information and Software Technology*, *51*, 809–814. doi:10.1016/j.infsof.2008.05.010

Hafner, M., Alam, M., & Breu, R. (2006, October 1-6). Towards a MOF/QVT-based domain architecture for model driven security. In O. Nierstrasz, J. Whittle, D. Harel, & G. Reggio (Eds.), *Proceedings of the 9th International Conference on Model Driven Engineering Languages and Systems*, Genova, Italy (LNCS 4199, pp. 275-290).

Copyright © 2011, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

Hafner, M., Breu, M., Breu, R., & Nowak, A. (2005). Modelling inter-organizational workflow security in a peer-to-peer environment. In *Proceedings of the IEEE International Conference on Web Services*.

Hafner, M., & Breu, R. (2009). *Security engineering for service-oriented architectures*. Berlin, Germany: Springer-Verlag.

Haug, T. H. (2007). *A systematic review of empirical research on model-driven development with UML*. Unpublished master's thesis, University of Oslo, Oslo, Norway.

Howard, M., & Lipner, S. (2006). *The security development lifecycle*. Sebastopol, CA: Microsoft Press.

Jensen, J., & Jaatun, M. G. (2011). Security in model driven development: A survey. In *Proceedings of the 5th International Workshop on Secure Software Engineering*.

Jürjens, J. (2005). *Secure systems development with UML*. New York, NY: Springer.

Jürjens, J., Schreck, J., & Bartmann, P. (2008). Model-based security analysis for mobile communications. In *Proceedings of the 30th International Conference on Software Engineering*.

Kasal, K., Heurix, J., & Neubauer, T. (2011, January 4-7). Model-driven development meets security: An evaluation of current approaches. In *Proceedings of the 44th Hawaii International Conference on Systems Science*.

Kitchenham, B. (2004). *Procedures for performing systematic reviews*. Staffordshire, UK: Keele University.

Kleppe, A. G., Warmer, J., & Bast, W. (2003). *MDA explained: The model driven architecture: Practice and promise*. Reading, MA: Addison-Wesley.

Lloyd, J., & Jürjens, J. (2009). Security analysis of a biometric authentication system using UMLsec and JML. In *Proceedings of the 12th International Conference on Model Driven Engineering Languages and Systems*.

McDermott, J. (2005). *Visual security protocol modeling*. Paper presented at the New Security Paradigms Workshop.

McGraw, G. (2006). *Software security: Building security*. Reading, MA: Addison-Wesley.

Moebius, N., Stenzel, K., Grandy, H., & Reif, W. (2009a). Model-driven code generation for secure smart card applications. In *Proceedings of the Australian Software Engineering Conference.*

Moebius, N., Stenzel, K., Grandy, H., & Reif, W. (2009b). SecureMDD: A model-driven development method for secure smart card applications. In *Proceedings of the International Conference on Availability, Reliability and Security*.

Moebius, N., Stenzel, K., & Reif, W. (2009). Generating formal specifications for security-critical applications - A model-driven approach. In *Proceedings of the ICSE Workshop on Software Engineering for Secure Systems* (pp. 68-74).

OMG. (2010). *Executive overview - Model driven architecture.* Retrieved September, 2011, from http://www.omg.org/mda/executive_overview.htm

OWASP. (2011). *Category: OWASP top ten project.* Retrieved from http://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project

Rodriguez, A., Fernandez-Medina, E., & Piattini, M. (2006). Security requirement with a UML 2.0 profile. In *Proceedings of the 1st International Conference on Availability, Reliability and Security*.

Rodriguez, A., Fernandez-Medina, E., & Piattini, M. (2006). Towards a UML 2.0 extension for the modeling of security requirements in business processes. In S. Fischer-Hübner, S. Furnell, & C. Lambrinoudakis (Eds.), *Proceedings of the 3rd International Conference on Trust and Privacy in Digital Business* (LNCS 4083, pp. 51-61).

Rodriguez, A., Fernandez-Medina, E., & Piattini, M. (2007). Towards CIM to PIM transformation: From secure business processes defined in BPMN to use-cases. In G. Alonso, P. Dadam, & M. Rosemann (Eds.), *Proceedings of the 5th International Conference on Business Process Management* (LNCS 4714, pp. 408-415).

Rodriguez, A., Fernandez-Medina, E., & Piattini, M. (2008). CIM to PIM transformation: A reality. In *Proceedings of the IFIP TC 8 WG 8.9 International Conference on Research and Practical Issues of Enterprise Information Systems II* (Vol. 255, pp. 1239-1249).

Soler, E. TruJillo, J., Fernandez-Medina, E., & Piattini, M. (2007a). Application of QVT for the development of secure data warehouses: A case study. In *Proceedings of the 2nd International Conference on Availability, Reliability and Security* (pp. 829-836).

Soler, E., Stefanov, V., Mazon, J.-N., Trujillo, J., Fernandez-Madina, E., & Piattini, M. (2008). Towards comprehensive requirement analysis for data warehouses: Considering security requirements. In *Proceedings of the 3rd International Conference on Availability, Reliability and Security* (pp. 104-111).

Copyright © 2011, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

Soler, E., Trujillo, J., Blanco, C., & Fernandez-Medina, E. (2009). Designing secure data warehouses by using MDA and QVT. *Journal of Universal Computer Science*, *15*(8), 1607–1641.

Soler, E., Trujillo, J., Fernandez-Medina, E., & Piattini, M. (2007b). A framework for the development of secure data warehouses based on MDA and QVT. In *Proceedings of the 2nd International Conference on Availability, Reliability and Security* (pp. 294-300).

Soler, E., Trujillo, J., Fernandez-Medina, E., & Piattini, M. (2007c). A set of QVT relations to transform PIM to PSM in the design of secure data warehouses. In *Proceedings of the 2nd International Conference on Availability, Reliability and Security* (pp. 644-654).

Tøndel, I. A., Jaatun, M. G., & Meland, P. H. (2008). Security requirements for the rest of us: A survey. *IEEE Software*, *25*(1), 20–27. doi:10.1109/MS.2008.19

Trujillo, J., Soler, E., Fernández-Medina, E., & Piattini, M. (2009a). An engineering process for developing secure data warehouses. *Information and Software Technology*, *51*, 1033–1051. doi:10.1016/j.infsof.2008.12.003

Trujillo, J., Soler, E., Fernández-Medina, E., & Piattini, M. (2009b). A UML 2.0 profile to define security requirements for data warehouses. *Computer Standards & Interfaces*, *31*(5), 969–983. doi:10.1016/j.csi.2008.09.040

Wyk, K. R. v., & McGraw, G. (2005). Bridging the gap between software development and information security. *IEEE Security and Privacy*, *3*, 75–79. doi:10.1109/MSP.2005.118

Jostein Jensen graduated from the Norwegian University of Science and Technology (NTNU) in 2007, and has since been employed as a research scientist at SINTEF ICT in Trondheim. He is currently pursuing his PhD at NTNU. His research interests include software security, security in Air Traffic Management, and federated identity management systems.

Martin Gilje Jaatun graduated from the Norwegian Institute of Technology (NTH) in 1992, and has been employed as a research scientist at SINTEF ICT in Trondheim since 2004. His research interests include software security "for the rest of us", information security in process control environments, and security in cloud computing.

Copyright © 2011, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.