An Empirical Study of the State of the Practice and Acceptance of Model-Driven Engineering in Four Industrial Cases

Parastoo Mohagheghi[1], Wasif Gilani[2], Alin Stefanescu[3], Miguel A. Fernandez[4]

[1] *SINTEF, Oslo, The Norwegian University of Science and Technology, Trondheim, Norway*

[2] *SAP Research Center Belfast, United Kingdom*

[3] *University of Pitesti, Dept. of Computer Science, Pitesti, Romania*

[4] *Ericsson, Valladolid, Spain*

Phone (corresponding author): (0047) 22067497

Fax (corresponding author): (0047) 22067350

Email (corresponding author): parastoo.mohagheghi@sintef.no

Abstract. Model-Driven Engineering (MDE) has been promoted for many years as a means for handling the complexity of software development by raising the abstraction level and automating labor-intensive and error-prone tasks. However, there is little empirical evidence of the acceptance of MDE in industry which is the subject of this paper. The goal of this empirical study was to investigate the state of the practice of applying MDE and factors considered as important for its adoption. The subjects were developers of four large companies participating in a research project. The collected data came from multiple sources and covered the results of tool evaluations, interviews, and a survey. Among the factors, we found perceived usefulness, ease of use and the maturity of the tools to be important determinants for the adoption of MDE. We also discuss challenges with adopting MDE and present suggestions on how to succeed with the adoption process.

Keywords: *model-driven engineering, empirical study, technology acceptance model, qualitative study, survey*

1. Introduction

Today's software systems are complex in nature. Their size has been continuously growing due to increased functionality and new requirements such as dynamicity and autonomy. Additionally, heterogeneity is becoming an ever greater concern as systems are built from several subsystems, may include legacy code or are distributed over multiple sites. Model-Driven Engineering (MDE) has been promoted for a long time as a means for handling the development of complex software systems. In this paper, MDE refers to all methodologies that are model-centric and which encourage efficient use of models during all stages of software development. MDE is built upon many of the successful techniques applied in software engineering; i.e. raising the abstraction level by hiding platform-specific details, taking advantage of models to improve understanding, developing domain-specific solutions to fit the needs of the domain, and taking advantage of transformations in order to automate repetitive work and improve software quality. The approach is promoted by the Object Management Group (OMG)[1], which is working on several standards for MDE and companies such as IBM, which is spearheading MDE through the Eclipse Foundation's support of tools like EMF[2] and GMF[3]. Other advanced tools are being developed both as commercial and free products.

While MDE has received considerable attention from some companies, others are still reluctant to adopt it. The purpose of this paper is to describe the state of practice of applying MDE and to investigate factors considered as important for its adoption. This study is performed in the context of four industrial partners involved in a large European Integrated Project (IP) supported by the European Commission. The MODELPLEX project[4] (MODELing solution for comPLEX software systems) ran from 2006 to 2010 and had 21 partners, among them 4 large companies with cases of complex systems to explore and develop using MDE techniques. The empirical data collected and analyzed in this study cover descriptions of requirements and evaluation criteria, the results of tool evaluations, interviews with developers and a survey conducted among the industrial partners. Some of the companies involved in this project were already applying MDE while others continue performing research before committing themselves to a new paradigm.

The remainder of the paper is organized as follows. Section 2 presents the research context and Section 3 presents related work on the motivations and challenges of applying MDE. Section 4 describes the research method and data collection procedures. Section 5 is an introduction to the Technology Acceptance Model (TAM) and its extensions. This model is used in this study for evaluating the factors perceived as important for adopting MDE. Sections 6 and 7 present the findings that emerged from the analysis of data, followed by discussions in Section 8. Section 9 provides conclusions and implications for future work.

2. The Research Context

The contributors to this research were four companies participating in the MODELPLEX project. The project had a total number of 21 partners; among them industrial partners, tool vendors, research organizations, academia and consultancy companies based in 8 countries. The project's main objective has been to create a coherent infrastructure for the application of MDE during the development and management of complex software systems within a variety of industrial domains. The industrial partners participating in the project were divisions of large-scale companies from four different domains including enterprise business applications, telecommunication, aerospace crisis management systems and geological systems. Research in the project was organized around four axes:

- *Model engineering capabilities* defined as developing methodologies and tools for modeling complex systems. This research area covered Domain-Specific Language (DSL) engineering, composition of models, traceability among models and during transformations to other models and code, and derivation of models from legacy systems.
- *Verification and Validation (V&V) capabilities*, which covered model verification engines, tools for performance checking, simulation frameworks, test generators and model debuggers.

---

[1] OMG, http://www.omg.org/
[2] Eclipse Modeling Framework (EMF), http://www.eclipse.org/modeling/emf/
[3] Eclipse Graphical Modeling Framework (GMF), http://www.eclipse.org/modeling/gmf/
[4] http://www.modelplex.org/

- *System management capabilities,* which covered managing and configuring models after deployment, especially for system of systems and highly distributed devices.
- *Common software architecture* for integrating the tools and technologies (generally referred to as solutions) in a platform, including a model repository.

The project aimed to be driven by the needs of industrial cases. At the beginning of the project, the industrial partners defined a set of requirements within the context of their case studies that reflected their business goals, objectives and needs regarding MDE technologies and tools. Tool vendors used these requirements as the basis for developing solutions. The project was also strongly focused on empirical evaluation of the results. In order to evaluate the solutions in a systematic way, all industrial partners defined case-specific *research questions* that reflected their goals for using MDE and their criteria for evaluation. Research questions were answered by performing empirical studies during the project such as experiments and internal surveys. In order to provide a common platform for evaluating MDE as a methodology, we took advantage of the Technology Acceptance Model (TAM) (Davis 1989; Davis et al. 1989) and some extensions of it for collecting developers' perceptions of factors that are assumed to be important for the usage of MDE and intentions for future usage. TAM is used as the model that includes factors which may explain the adoption of a methodology.

3. Motivations and Challenges for Adopting MDE in Industry

In this section, we present related work on the motivations for adopting MDE in industry and some reported challenges.

The results of a systematic review on applying MDE in industry are presented in (Mohagheghi and Dehlen 2008). This review covered 25 papers published from 2000 to 2007 in a number of major journals and events related to MDE. Regarding motivations for applying MDE, we can group them in two:

1. *To improve productivity and shorten development time* by increasing automation (generating code and other artifacts from models) and promoting reuse. For example the motivation can be the provision of a common framework for software development across the company and lifecycle, or the development of Platform Independent Models (PIMs) that have a long lifespan and may be ported to multiple platforms.
2. *To improve software quality* by earlier detection of bugs, model-based simulation and testing, maintaining the architecture consistency from analysis to implementation, improved communication and information sharing among stakeholders, and traceability throughout software development artifacts.

There was little empirical evidence reported in the papers. The quantitative data were related to the effort spent on developing applications, and effort spent on inspections and other activities for detecting defects. Some experiences regarding tool support were also reported.

In another MDE adoption study, the costs and benefits of MDE in the car industry were studied (Kirstan and Zimmermann 2010). In this study, 12 interviews were conducted with car producers, their suppliers and consulting companies. Motivations to use MDE were development cost savings, the fear of missing the next generation development technology, competitive pressure, and the experience that many innovative functions can only be developed via model-based technologies. The companies hoped to achieve the following advantages because of MDE: a higher degree of automation, the possibility of detecting errors earlier, better communication with colleagues and reduced effort for reusing functions in other car lines. Negative experiences were also reported such as high dependence on tool vendors, bugs in the tools and the need for process redesign. Some also mentioned that they cannot trust code generators to produce safety-critical and efficient code with the same quality as hand-written code, while others were satisfied with the quality of generated code.

Results of a survey among industrial participants, reported in (Staron 2006), showed that the availability of tools was perceived as the most influential factor for the adoption of MDE. Tool performance was also emphasized by participants in another research project as a major barrier to MDE adoption since they experienced problems with instability of the tools and their integration (MODELWARE 2006).

Single studies have also been published since 2007 which report on the motivations and benefits of applying the MDE paradigm, especially in the series of workshops on domain engineering[5]. Furthermore, at the Future of Software Engineering Research sessions held in the International Conference on Software Engineering (ICSE) in 2007 (France and Rumpe 2007), the main challenges of MDE were outlined as: modeling languages providing support for creating and using problem-level abstractions; challenges regarding separation of concerns when systems are modeled using multiple, overlapping viewpoints or languages; and model manipulation and management issues such as maintaining consistency between models and traceability among model elements. Several of these challenges were also observed by the participants in this empirical study, as discussed later in the paper.

4. Research Method and Data Collection

This study uses *qualitative research* where the researcher often collects open-ended, emerging data with the primary intent of developing themes and theories from the data. There are different strategies for performing qualitative research such as ethnographies, grounded theory, case studies and narrative research (Creswell 2002). The research applied in this project is an *exploratory case study* (Yin 2002). Multiple types of data are collected and analyzed to study the impact of MDE in industrial cases. As an *empirical study,* the intervention (here MDE) is being evaluated and there is no single set of outcomes. As previously mentioned, the research participants for this study are industry partners in a research project so the selection of cases was by convenience rather than any other criteria.
The empirical studies done in the project (2006-2010) may be divided into two groups:

- *Context-specific research* designed and performed by the companies, addressing their objectives of applying MDE techniques.
- *MDE acceptance research* designed and conducted by a researcher external to the companies and addressing research questions regarding the acceptance of MDE by all of the companies.

The timeline of the empirical studies performed by the companies is as follows. The companies defined their scenarios for applying MDE, requirements and expected outcomes in form of research questions in the first year of the project (2006-2007). The scope of these research questions depended on the companies' interests in MDE and the possible benefits of model-based testing, code generation from models, benefits of a DSL or any other MDE technique. They later performed different empirical studies in their contexts using the tools provided by technology providers (sometimes combined with internal tools) in 2007-2009 and reported their findings as answers to the research questions in an internal evaluation report in 2010. The empirical studies were prototyping, comparisons with other tools, experiments or even internal surveys. Each company participated in the project with a team of developers that were involved in applying MDE. The evaluation was sometimes done by these teams and sometimes involved larger groups of people within their companies.
The MDE acceptance research consisted of two parts: 1) semi-structured interviews with the development teams of companies performed by the external researcher in June 2008; 2) an on-line survey made available to the development teams in January 2010 where responses were collected and analyzed in February 2010. The interview and survey were designed according to the Technology Acceptance Model (TAM) that is presented in the next section. The findings of the interviews and survey are the subject of this paper. However, in order to explain some of the findings, we also take advantage of some of the observations from empirical research performed by the companies.
The data analyzed and used in the paper thus cover:

1. Requirements and criteria for evaluating MDE as defined by companies in their research questions. These criteria are presented in (Mohagheghi 2010).
2. Observations from internal studies and tool evaluations performed by the companies.
3. Semi-structured interviews conducted in 2008 with questions on the usage of MDE before the project, and the companies' plans to experiment with the project tools. These interviews were performed with one team of two to three developers from each company who were involved in the project. The interview was designed and conducted by a researcher who was not involved in any of the case studies. Each interview lasted between 25 to 45 minutes and was

[5] See http://domainengineering.org/ and http://www.dsmforum.org/

recorded and transcribed in full. The text and conclusions were sent to the interviewees for comments and corrections. The full list of questions is provided in Appendix I.

4. An on-line survey with questions similar to previous studies using TAM as described in the next section. The survey was conducted in 2010 using LimeSurvey[6] and reflects the status and opinions of the practitioners involved. The questionnaire used for the on-line survey is provided in Appendix II.

In January 2010, we distributed the questionnaire among all developers involved in the industrial cases. While we received responses from two companies completed by individual developers (six responses), the other two companies were reluctant to provide us individual responses and completed the questionnaire as groups in order to follow recommendations of their companies regarding surveys, and also to have consensus regarding current and future use intentions. The number of responses to this part of the study is too low to generate any statistics. The results are, however, important when interpreted in the context of case studies, especially since there have been few empirical studies performed on this scale and in large companies. We therefore summarized the survey findings per development team and analyzed them in relation to other data collected from context-specific studies and semi-structured interviews (which were also summarized per development team). The conclusions were validated by the team answering the questionnaire.

The units of study in the interviews and the on-line survey are the development teams involved in the project. The companies are large multi-nationals and the development method varies between different departments and even from one project to another. The practitioners answered the questions based on their experience of the projects they were involved in and the findings are not generalized to other projects within the companies. However, we consider the findings as important since they are based on the opinions of developers using or experimenting with MDE for at least the four-year duration of this project, and some for even longer. Moreover, these developers were among the most knowledgeable in MDE and usually had a good overview of other MDE efforts in their companies. Their interest in making MDE work in their contexts may be considered as a bias for some findings. However, they provide expert opinion on the subject and discuss impediments and challenges.

In the next section, we present the technology acceptance model used in this research.

5. The Technology Acceptance Model

5.1. General Introduction

Davis et al. (1989) introduced TAM to explain the adoption of information technology applications in organizations. The original model explains users' intention to use a new technology (or system) through two beliefs: *perceived usefulness,* which is the extent to which a person thinks that using a system will enhance his or her job performance and *perceived ease of use,* which is the extent to which the person thinks that using the system will be easy. In later extensions of TAM, additional factors are added, such as *subjective norm*, which is defined as the degree to which a person thinks that others who are important to the person think that he or she should use the system.

Riemenschneider et al. (2002) have compared five theoretical models for explaining acceptance of methodologies: TAM, TAM2 (an extension of TAM), Perceived Characteristics of Innovating (PCI), Theory of Planned Behavior (TPB) and Model of Personal Computer Utilization (MPCU). The five models have common constructs (or factors) as presented in Table 1. *Facilitating conditions* in MPCU refers to factors in the environment that make using a system easier. There are also factors that are unique to the models. PCI includes, for example, the factor *compatibility*, which refers to the degree to which an innovation is perceived as being consistent with existing values, needs and past experience of potential adopters. Other unique factors in these models are not considered as important for this study.

Table 1. Comparison of common factors in five models (Riemenschneider et al. 2002)

Construct	**TAM**	**TAM2**	**PCI**	**TPB**	**MPCU**
Usefulness	Usefulness	Usefulness	Relative	Attitude	Job fit

[6] http://www.limesurvey.org/

Construct	TAM	TAM2	PCI	TPB	MPCU
			advantage		
Ease of use	Ease of use	Ease of use	Complexity		Complexity
Subjective norm		Subjective norm		Subjective norm	Social factors
Voluntariness		Voluntariness	Voluntariness		
Perceived Behavioral Control (PBC)				PBC-External	Facilitating conditions

For measuring the relation between the above factors and the acceptance of a technology or system, a questionnaire is often used and intercorrelations, means, standard deviations and Cronbach's alphas[7] are calculated. The question is whether these factors as independent variables can explain intentions for using a technology as the dependent variable.

Dybå et al. (2004) have selected a variety of factors, including some belonging to TAM2, and developed a model for evaluating the acceptance of an Electronic Process Guide (EPG) in a company. Their model is depicted in Figure 1. Due to the voluntary nature of using EPG, they dropped voluntariness, while they added *organizational support* to the model, which is the degree to which a method is supported by an organization. Also they made a distinction between current use (since the system is already in use) and future use intentions. In Figure 1, the factors assumed to be important for accepting EPG (the independent factors) are shown on the left-hand side while the dependent factors (current system usage and future use intentions) are shown in a box on the right-hand side. The arrows indicate the impact of the independent factors on the dependent factors.

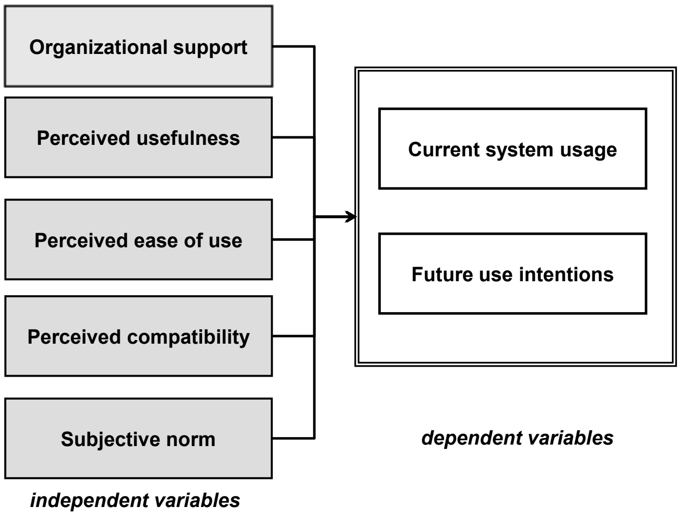

Fig. 1. The conceptual model for evaluating EPG used in (Dybå et al. 2004)

They conducted a survey of all prospective users of EPG (120 employees) and found that, consistent with prior research, perceived usefulness is a strong and significant factor in explaining current usage and future use intentions, while the other independent factors did not have any significant impact on these two. However, organizational support, perceived ease of use and subjective norm were significant determinants of perceived usefulness. EPG is also the subject of a qualitative study (Moe and Dybå 2006) where semi-structured interviews were used to collect opinions instead of a survey.

Walderhaug et al. (2008) used TAM in the MPOWER project to evaluate the use of MDE in the healthcare domain. They dropped organizational support since this was a research project while an additional factor on *tool performance* was added since they assumed that the adoption of MDE depends strongly on the availability of suitable tools. In this study, tool performance specifically targeted business analysis features, traceability and implementation automation. 16 developers from 4 European countries completed a survey and their findings suggested that perceived usefulness and ease of use were the most important factors for the adoption of MDE. No

[7] Cronbach's alpha is a measure of intercorrelation among test items. Intercorrelations are maximized when all items measure the same construct.

significant relations between future use intentions and tool performance or subjective norm were found.

5.2. TAM as Used in this Study

As emphasized by Creswell (2002), qualitative research is often exploratory and is useful when the researcher does not know the important variables to examine. In this case we chose to study some variables that are proposed in variations of TAM for the purpose of answering what developers believe to be important for accepting MDE and whether the companies involved in this project intend to use MDE in their future work. TAM was selected since it has been used in similar studies (as described in the previous section) and has been evaluated as useful in describing factors that influence the acceptance of a technology. The context-specific research performed by the companies provided interesting results, while the studies differed in objectives and research questions. This motivated us to use TAM as a common framework for collecting opinions.
We chose the following as independent factors:

- *Perceived usefulness,* which is defined as the extent to which a participant thinks that applying MDE will enhance his or her job performance.
- *Perceived ease of use,* which is defined as the extent to which a participants thinks that applying MDE will be easy.
- *Perceived compatibility*, which is defined as the extent to which the MDE method is perceived as compatible with existing development methods and practices in these contexts.
- *Subjective norm*, which is defined as the extent to which a developer thinks that others who are important to him/her think that he or she should use MDE.
- *Perceived maturity of MDE tools* developed in the project. This factor is included based on the assumption that the maturity and performance of tools impact the acceptance of MDE as a method, similar to (Walderhaug et al. 2008).

We chose not to evaluate *organizational support* since the companies involved are participating voluntarily in this research project and the project is supported by their organizations.

The independent variables in the model are *Current usage* (the extent companies are using MDE) and *Future use intentions* of MDE defined as the extent to which the companies intend to use MDE in their future work. However, we think that the process of MDE acceptance is actually a complex one where the current usage itself acts as an intermediate factor affecting *Future use intentions*. Fig. 2 depicts the model that was applied in this research.

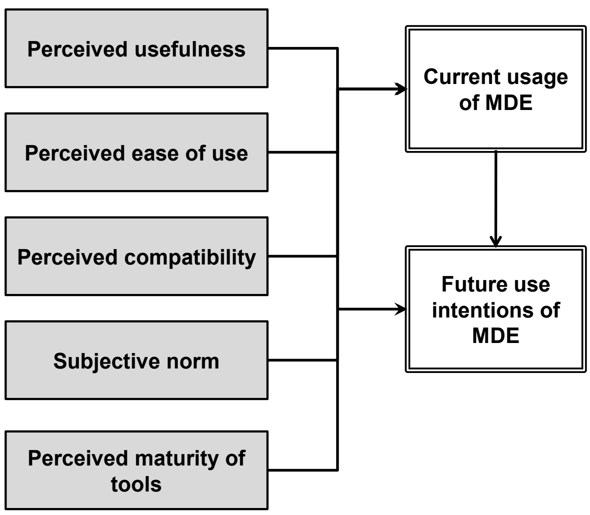

Fig. 2. The updated TAM used in this study

The survey (detailed in Appendix II) was based on the above model and included the following categories of questions:

1. *Participant background* including the level of expertise on MDE and the number of years using MDE.
2. *Current usage of MDE* with the question "How would you describe the current usage of the MDE approach in different projects within your department?" The current usage was measured

on a scale from: “not used at all”, “used on an experimental basis”, “used on a regular basis by few people or projects”, “used on a regular basis by most people or projects”, and “used on a regular basis by all people or projects”.

3. *Perceived usefulness* with six statements such as “I find the MDE approach useful in my job”.
4. *Perceived ease of use* with five statements such as “Using the MDE approach does not require a lot of mental effort”.
5. *Perceived compatibility* with four statements such as “Using the MDE approach fits well with the way I work”.
6. *Subjective norm* with three statements such as “People who influence my work think I should use the MDE approach”.
7. *Perceived maturity of MDE tools* with six statements such as “The MDE tools I use have acceptable performance”.
8. *Future use intentions of MDE* with four statements such as “I would like to use the MDE approach in the future for my work”.
9. An open-ended question “Based on your experience, what are the main advantages and disadvantages of using the MDE approach”.

For the statements, the scale was: “strongly disagree”, “disagree”, “neither agree nor disagree”, “agree”, and “strongly agree”.
In addition to the statements on MDE as a methodology, the questionnaire covered questions specific to MDE tools developed in the project, and some common platforms and technologies. This section covered:

10. *Current usage of MDE tools*: “Which of the following tools have you been using or experimenting with in the project? A list of 22 tools and technologies developed in the project in addition to the Eclipse platform, UML profiles and Domain-Specific Languages was provided. The scale was the same as for *Current usage of MDE*.
11. *Future usage intentions of MDE tools*: “How would you describe your intentions regarding future use of the following tools?” The same list of tools as above was used. The scale was: “yes”, “probably yes”, “experimenting”, “don’t use it and don’t think I will do”, “no”, and “don’t know”.
12. *Scores for the tools* that the companies have used or experimented with. The scale was from 0 to 4, where 0 indicated “very dissatisfied” and 4 indicated “very satisfied”. Five factors were included in evaluating the tools: functionality, ease of use, compatibility, performance and reliability, and total impression. These factors were identified from research questions studied during the project.

The companies participating in the study had different levels of experience with MDE and the findings reflect the diversity between companies and contexts.

6. Current Usage of MDE in the Surveyed Companies

In this section we describe the state of the practice of applying MDE. As mentioned, the companies are large industrial organizations and the answers reflected the state of practice as observed by the developers in their departments, and also their awareness of MDE initiatives from other groups. The companies are anonymous as requested by them.
In the interviews, we discovered that all companies used models in their development, especially during analysis and design phases. However, the practices varied significantly regarding the use of models for other purposes such as analysis and documentation. Since most of the tools developed in the project are based on the Eclipse platform, we asked also companies about their experience with Eclipse.

6.1. Company *A*

Company *A* characterized itself as an advanced user of MDE that worked with it on a regular basis. They used models for generating code, user interfaces and even for generating test cases. They also had experience with domain-specific modeling solutions and had internal tools based on these. Models were integrated into their development process (which is a proprietary one) and they even used a model repository. The Eclipse platform, which was used in this project as the integration platform, was widely used in company *A*.

In this project, the focus of company *A* has been applying MDE to new activities, such as performance engineering, and extending their application of MDE into areas where it has not been applied before, such as model-based testing.

6.2. Company *B*

Company *B* used MDE on an experimental basis in different research projects. Their development process has so far been code-centric where models have been mainly used for documentation purposes. They had no experience with domain-specific modeling and UML profiles before joining this project. Regarding Eclipse, it has been used as a development tool for Java programming. Company *B* had a proprietary development process, which is mainly focused on the managerial aspects of projects such as deadlines, tasks and activity management. It was not oriented towards any software development process, such as agile, or any software development methodology, such as model-based or object-oriented development.
The main purpose of participating in this project has been gaining experience with MDE and evaluating whether it can be used in their context to improve their development and maintenance processes. Their focus was specifically on developing domain-specific languages and tools that improve communication among teams, allow easier configuration and reuse, and may be used in generating artifacts based on models.

6.3. Company *C*

Company *C* characterized itself as experienced with modeling, using UML profiles and generating artifacts from models. When it comes to integrating MDE in their development process, the practice is project specific. They had a proprietary development process where the core was defined, and the details were adapted to the type of product and the customer demands, and may also include modeling and MDE techniques. The development team participating in the project had used MDE in a few projects.
The company's focus in this research project had been applying MDE for modeling and managing complex systems. Some additional activities were also performed such as modeling security aspects and weaving these with other models.

6.3. Company *D*

Company *D* characterized itself as experienced with MDE having experimented with it in several research projects. However, their current development process is code-centric and models are mainly used for the initial analysis and design of systems. In this process, they also used Eclipse as a development tool for Java programming. In company *D*, Scrum[8] (an agile methodology) was the day-to-day development process while the Rational Unified Process (RUP)[9] was used in the bigger life cycle process for modeling in the initial phases of software development.
Their main purpose of participating in the project was to develop domain-specific solutions for modeling their context and especially model-based simulation of the performance of the system and the external world in order to improve their design.

7. Survey and Interview Findings

This section presents the findings of the survey and the semi-structured interviews. Average scores of the responses are given in Table 2. The findings are grouped using the independent factors defined in Section 5.2.

7.1. Participant Background

The participants in the survey characterized themselves as moderate to expert MDE users. Their experience with MDE varied from 3.5 years to more than 10 years.

---

[8] http://scrummethodology.com/
[9] http://www-01.ibm.com/software/awdtools/rup/

7.2. Perceived Usefulness

Perceived usefulness is the degree to which a person believes that using the MDE approach will enhance their job performance. The survey findings are:

- Most consensus was on the statement "I find the MDE approach useful in my job", where almost everyone answered with "agree".
- On the statements "Using the MDE approach enhances the quality of my job" and "Using the MDE approach increases my productivity", the responses were "neither agree nor disagree" or "agree", with slightly more of the second category.
- Other statements such as "Using the MDE approach makes it easier to do my job" and "Using the MDE approach improves my job performance" were mostly replied with "neither agree nor disagree".

No one strongly agreed or disagreed with the above statements and there were no significant differences in the findings per company, except for one question. When asking "Overall the advantages of using the MDE approach outweighs the disadvantages", company *D* disagreed while all others agreed. Company *D* believes that customizing a development process for MDE requires a lot of effort, which does not pay off for small projects, and the current tools are not mature enough for larger projects.

7.3. Perceived Ease of Use

Five statements in the survey covered ease of use:

- "Learning to use the MDE approach was easy for me" was replied with "neither agree nor disagree" by all four companies. Regarding the statement "I find the MDE approach easy to use", only company *A* agreed while the others replied with "neither agree nor disagree".
- Other statements; i.e., "I think the MDE approach is clear and understandable", "Using the MDE approach does not require a lot of mental" effort and "The MDE approach is not cumbersome to use" were answered by "neither agree nor disagree" or "agree" by three companies, while company *D* disagreed.

The overall impression is that learning MDE is not easy. However, company *A,* which was using MDE in most projects was most positive regarding ease of use, and company *D,* with experimental usage of MDE, was most negative. This may be an indication of the steep learning curve which is mentioned in the interviews by statements such as[10]:

> *C: It is difficult to find trained people and it is difficult to train people who haven't had this kind of background.*

> *C: Tools are really too much oriented towards the engineer. We cannot develop a business process model in an easy and understandable way for the customer.*

> *B: The complexity is high, not because the tool user interface is not good but because UML is complex per se.*

The survey did not ask the subject to compare MDE with other development approaches.

7.4. Perceived Compatibility

For each company, several tools had to be integrated into a tool chain that often consisted of tools already in use and new tools developed within the project. In addition to common modeling tools, the companies also took advantage of proprietary modeling languages and tools developed around them. Integrating tools required transforming models or using scripts. The effort required for integration made the companies emphasize the usage or conformance to standard languages in their requirements to the tool vendors.

In the survey, four statements covered perceived compatibility. Company *D* disagreed with all of the statements and noted that the compatibility of the MDE approach with existing practices is an impediment for practicing it on a wide scale in their context. At the moment, this company used MDE only on an experimental basis. In the interview, one representative of this company said:

[10] These items are direct quotes.

D: We are going to use models and transformations much more in the future, so there need to be well-defined standards because otherwise you are risking wasting your effort. Two years later there might not be any support from the tool providers available. I guess current standards are not going to be the same in two years time. So this is a very risky business so far.

Companies *A*, *B* and *C* replied with "strongly agree" or "agree" to the statements on compatibility, except for one statement. When asking "Using the MDE approach is compatible with all aspects of my work", company *C* disagreed. However, companies *A* and *B* have also faced problems in integration. In the interviews, developers of company *B* said that there are interoperability problems between the modeling tool that they are using and some project tools. Company *A* had to develop adapters to transform their proprietary models into UML models in order to import them in other tools.

7.5. Subjective Norm

Subjective norm is the degree to which software developers think that others who are important to them think they should use a method or tool. The three statements were mostly replied with "neither agree nor disagree", while company *A,* which is already using MDE in most projects, had highest agreements. We think that developers perceived their co-workers and others who are important to them as open-minded regarding applying MDE as long as there are benefits to show. Subjective norm was only covered in the survey.

7.6. Perceived Maturity of MDE Tools

Perceived maturity is defined as the degree to which tools are perceived as mature and suitable for the tasks in hand. We discuss first interview findings, where we asked developers about their opinions regarding the maturity of MDE tools. Company *A,* with a wider usage of MDE, found MDE tools most mature, partly because they have developed several internal tools. Company *C* found some tools mature and others not, while companies *B* and *D,* who are experimenting with MDE did not find the MDE tools mature enough.
The survey questions on various aspects of MDE tools produced more insight into the reasons for finding the MDE tools immature:

- On the statement "The MDE tools I use are suitable for both small and large projects", only company *A* agreed while all of the others disagreed. In this research, projects with less than 15 developers are considered as small, while projects that involve several development teams and considerably more developers are considered as large.
- There was no consensus on the statement "The MDE tools I use provide the functionality I need" and the responses varied from "disagree" to "agree".
- The statement "The MDE tools I use have acceptable performance" was answered with either "agree" or "neither agree nor disagree". The same for the statement "Using MDE tools improve the way I develop software".

We conclude that the performance of tools was perceived as acceptable, although there were no strong agreements. However, their scalability to large projects was a major concern.

7.7. Intentions for Future Usage of MDE

Four statements in the survey covered this aspect:

1. On the two statements "I would like to use MDE in the future" and "I intend to use the MDE approach in the future for my work", all companies answered with "agree".

- Regarding the statement "I intend to increase my use of the MDE approach for work in the future", only company *A* agreed while others replied with "neither agree nor disagree".
- There was also a negative statement saying that "Given a choice, I would prefer *not* to use the MDE approach for work in the future" where companies answered with "disagree" or "neither agree nor disagree".

We conclude that despite problems with the ease of use and scalability of tools, the development teams participating in the survey generally thought that they would like to use MDE in their future work or at least they were not reluctant to use it. Company *A* which was most advanced in using MDE was the most determined to increase their usage of MDE.

Table 2 summarizes the scores given to the questions in the survey for the four companies. The sample is too small to generate statistical conclusions, although the scores are useful for understanding the findings. The scale was from 0 to 4 where 0 indicates strong disagreement and 4 indicates strong agreement. Companies *A* and *B* had a higher opinion regarding MDE and were also more positive regarding future use. The difference is that company *A* used MDE in real projects, while company *B* did not. Company *C* scores are lower regarding maturity of tools and compatibility, while company *D* was most negative regarding all factors but still scored high regarding future use intentions. This may be explained by interview findings where company *D* mentioned that they found MDE useful for specific activities, such as simulation and analyzing the design at an early stage, although they did not find MDE applicable in the whole development lifecycle. Opinions regarding tools developed in the project and MDE techniques were generally positive, which is also important for describing future use intentions.

Table 2. Average scores given by companies to the questions in the survey

Company	Perceived usefulness	Perceived ease of use	Perceived compatibility	Subjective norm	Perceived maturity of tools	Future use intentions
A; used in most projects	3.0	2.6	2.5	2.7	2.7	3.0
B; experimental user	2.5	2.0	3.1	1.8	2.3	3.0
C; used in few projects	2.75	2.1	2.3	1.8	2.0	2.7
D; experimental user	2.0	1.8	1.0	2.0	1.2	2.7

7.8. Motivations and Obstacles for Adopting MDE

Here we use the findings of interviews and the open-ended question in the survey to discuss motivations, facilitating factors or inhibiting factors for adopting MDE in industry.
Developers from the companies who were using MDE in industrial projects (companies *A* and *C*) mentioned *reduced complexity* in development, *improved communication* with non-technical experts and *consistency* between artifacts as their main motivations for using MDE. They said:

> *A: Different people are involved such as business analysts that can use models for communication. Our development includes domain experts who would prefer domain-specific models.*

> *C: Modeling helps with getting a better overview by providing higher levels of abstraction. It is much easier to enter the project when you have models so you can understand what the application is doing.*

> *C: Quality would be improved when we have clear models of the architecture, and we can maintain consistency between the different artifacts that are produced.*

Unlike the findings presented in (Mohagheghi and Dehlen 2008), increased productivity was not mentioned as a goal in the interviews and neither was it measured in the context-specific empirical studies. There are several reasons to that. Firstly, it is difficult to show any productivity improvement in single projects. Secondly, applying the MDE approach requires spending effort on activities such as modeling, developing transformations and customizing tools where little information is collected regarding effort. Thirdly, companies considered most research in the project as exploratory. However, as mentioned in the interviews, companies foresee long-term improvements in productivity if the approach is applied in multiple projects. For example, developers from companies *B*, *C* and *D* mentioned that there may be *long-term productivity improvements* because of automating repetitive tasks, especially for product lines or when a company runs similar projects:

B: Applying a new methodology on a single project introduces high risk. A product line company has different iterations to enrich their products and they gradually may go for an MDE approach.

D: For medium-size or small projects I think that it gives you more work that it takes away. Perhaps for very big projects it could save you in the long term.

A: Application of MDE concepts enabled us to focus on the creative tasks of development rather than repetitive coding. The only place where a significant amount of coding effort was required was for the integration of tools into the existing tool infrastructure.

We also asked the interviewees "if you are not using MDE, why not?". Companies *B* and *D* who were using MDE only on an experimental basis had similar opinions:

B: Why that doesn't get out of the research environment and goes to production? Perhaps it has to do with the perceived maturity of the field by research teams.

B: We don't think it is beneficial to use it for small projects and we don't think it is mature enough for large projects.

B: Managing all these artifacts from models to code may add more complexity than anticipated at first.

D: It is because the whole approach is not mature, the tools are not mature enough, it is unclear how to apply it on a larger scale, and also that most of the developers are not convinced yet that this is better than the standard approach.

Regarding disadvantages of MDE, it was stated by companies *B* and *D* that *integrating tools* requires significant effort. The developers in company *C* also mentioned that the *tool chain* is broken in many places, and depending on the situation, they have to develop workarounds. Also, the *lack of well-trained people* in the domain was mentioned as an impediment by companies *B* and *D*.
In the survey, we had a statement that "Overall the advantages of using the MDE approach outweigh the disadvantages". Three of the companies agreed, while company *D* disagreed. One representative of company *D* had an interesting statement in the interview on the lack of observable benefits:

D: I think that models are fine, visual models are good, I think models can also be used to generate code or other artifacts in certain areas, for niche areas, for example for database mapping, for communication, and for maybe a little bit of testing. But the model-driven approach as advertised, that models are the core and code is secondary, is not proven at all today.

Some research participants have provided suggestions regarding how to succeed with the MDE adoption. They suggest:

- Identification of the areas where MDE can be perceived as useful. For example, solving business critical problems.
- Improving the user experience of tools by improving their Graphical User Interfaces (GUIs), creating documentation, examples and checklists.
- Developing a model on ROI and cost-benefit analysis and using it to convince developers and decision makers.
- Investigating the potential for reuse of solutions in other projects or products and implementing functionality for easy tailoring to different projects.
- Use of domain knowledge and company practices, by involving domain experts and customizing solutions for the specific domain.
- Evaluation of compatibility of tools with other tools and existing development processes, and the implementation of functionality that is needed for integration.
- Planning a gradual transition and involving users in the evaluation.

7.9. Usage of Project Solutions

In the interviews, we asked companies about the MDE tools and techniques they were using or planning to use during the project. These plans were modified during the project based on the availability of tools and the provided functionality, effort required to evaluate a tool and the changing priority of companies. For example, it showed that extracting models from legacy code in order to integrate legacy systems in a model-driven development chain required a lot of collaboration between industry and tool developers, and the companies were required to open their source code to external stakeholders. Tools developed for reverse engineering were therefore not evaluated on real industrial cases, but only on toy examples. However, several areas of the project's research received considerable attention by industrial users:

- Model composition tools, such as Reuseware composition framework[11], were used to develop, relate, and integrate models on several abstraction levels.
- Simulation solutions (both commercial tools and those developed by companies for their domain) were used to simulate external environments and business scenarios to help decision making.
- Domain-specific modeling solutions were developed in all four industrial contexts. For modeling, generating artifacts from models, and model-based simulation and testing.

Most project tools are Eclipse-based and the companies had previous experience with Eclipse, both as a Java programming platform, and as a platform for developing plug-ins and MDE tools. The overall impression of Eclipse was positive and the platform received the highest scores by survey participants (as given in Appendix III). However, Eclipse GMF for developing DSLs was perceived as complex and industry needs a more user-friendly and intuitive development platform, as also discussed in (Evans et al. 2009). Tool support for developing UML profiles is better while the complexity lies in the meta-model of UML. Developing a language suitable for the domain that is easy to use and at the right abstraction level is a challenge that requires several iterations and high language expertise. Although all companies have indicated that they will use DSLs or UML profiles in the future, they are also aware of the challenges and pitfalls.

We asked developers participating in the survey to assign scores to MDE tools developed in the project across five dimensions: 1) functionality; 2) ease of use, 3) compatibility; 4) performance and reliability; and 5) total impression. The results are presented in Appendix III for tools that were used by several companies in the project. Scores for Eclipse, DSLs and UML profiles are also given. Most tools received acceptable scores (higher than 2 from a total of 4), except for compatibility, where some tools received lower scores than considered acceptable. There may be a bias here since most of the tools are developed during the project or are improved to meet the requirements of industrial users. Regarding intentions for future usage of MDE tools, model repository tools, and tools for model-based simulation and verification, are those that have received the most positive responses, and some are already integrated in the companies' development chains.

8. Summary and Discussion

This study puts together data from internal empirical studies, interviews and a survey to investigate: a) the state of the practice and adoption of MDE in the context of four large industrial companies developing complex software systems; and b) to answer the question of which factors are considered as important for the adoption of MDE by the development teams of these companies. The findings are summarized below and compared with the findings of other research. Threats to validity are discussed as well.

8.1. MDE Acceptance Factors

The factors *usefulness*, *ease of use*, *compatibility, subjective norm* and *tool maturity* were selected for collecting opinions of development teams regarding MDE. Analyzing the interviews revealed an additional factor perceived as important for accepting MDE (i.e., the long-term Return-On-Investment (ROI) and cost-effectiveness of the approach). We discuss the factors in detail below.

Usefulness. The developers found MDE useful for doing their jobs although there were no strong agreements with the statements on usefulness. Developers from companies that were already using MDE were more positive regarding the usefulness of the approach. As reflected in the interviews, MDE was neither accepted nor rejected as a whole but found useful in some cases and activities. Usefulness is clearly a significant determinant to use the MDE paradigm and tools. Therefore it was the subject of several research questions in the companies' evaluation plans that were

[11] http://www.reuseware.org/index.php/Reuseware

discussed in Section 2. Examples of measures defined for usefulness are: the suitability of a DSL or UML profile for the problem; code-generation ratio from models; the quality of the generated artifacts; and the correctness of simulation models. Walderhaug et al. (2008) also found usefulness as an important factor for the adoption of MDE.

Ease of use. The findings of the survey and interviews indicated that ease of use is a significant factor in adoption, while on the other hand, the complexity of solutions and lack of expertise are considered as obstacles. Developers from companies who were already applying MDE found it easier to use than those who were experimenting with it. One reason is that MDE tools were better integrated in their development chains. Some participants had serious concerns that models actually add to the complexity of software development, partly due to the comprehension problems related to modeling languages. This added complexity must be compensated by having intuitive modeling languages and tools, and reducing the complexity of software development via code generation. Availability of tutorials and hands-on experience were also mentioned in the interviews as important for facilitating the adoption. Ease of use was also found as an important factor in the research presented in (Walderhaug et al. 2008).

Compatibility. Compatibility with existing tools and development practices is perceived as an important factor for adopting MDE since companies have expressed several requirements regarding compatibility of tools with their internal tools. The survey findings indicate that, except for company *D*, the others found MDE compatible with the way they work. The experience of the project shows that companies often develop adapters or find workarounds to solve the compatibility problems and do not consider them as major obstacles. Compliance with standards or existing platforms and meta-models was mentioned frequently in the requirements of companies in order to avoid dependence on tool vendors and achieve better integration. However, the participants knew that standards and platforms often evolve and new ones are emerging. One lesson learned in the project was to develop solutions around industry accepted platforms, such as Eclipse, and adapt other tools to these environments.

Maturity of tools and techniques. Immaturity of tools was mentioned as a serious barrier to adoption. Companies were generally satisfied with the performance of tools while scalability to large projects was a concern. Features, such as multi-user support, version control, ease of change and preservation of model properties in transformations, were required for developing large-scale projects. Overall, the common perception is that user friendliness and technological maturity of MDE tools for large projects are not yet at a reasonable level, especially among the companies who are experimental users. Kirstan and Zimmermann (2010) write that bugs in the tools led to negative user experiences. In this project, bugs were continuously reported to tool providers and fixed.

Subjective norm. The participants generally answered that they did not perceive the opinions of their colleagues or others who influence their job either in favor or disfavor of MDE. Subjective norm was not found as an important factor for future usage in (Walderhaug et al. 2008) either. MDE is no longer considered as a “hype”, while companies do not want to miss the advantages of this development technology, as stated by (Kirstan and Zimmermann 2010).

ROI and cost-effectiveness. Although ROI and cost-effectiveness are not included in TAM or similar models, the interviewees mentioned that solutions must pay off by reducing some costs or complexity of software development and being reusable in several projects. This was especially emphasized by the experimental users. Companies were concerned about effort required for developing transformations, cost of adopting a solution to a new project and effort required to set up a tool chain. An important aspect mentioned by interviewees was maintenance costs such as updating a domain-specific modeling language when the domain evolves or reusing code generators.

In software development, human resources are the main cost factor and companies were concerned about the cost of training their personnel to apply a new methodology. User friendliness of tools and their clarity are therefore crucial related to the costs, as reflected in the requirements. Participants in this research also emphasized the importance of reusing the artifacts or tools in multiple projects in order to compensate for the high initial cost of developing tools and training human resources. Similar conclusions were made by Kirstan and Zimmermann (2010) based on interviews with companies in the car industry where they reported that applying the paradigm alone does not lead to cost and time savings without reuse. An implication for our future research is to focus more on the cost-benefit view when studying success or failure of the methodology, and define appropriate measures to assess whether criteria regarding ROI have been met.

8.2. Current and Future Usage

Regarding current usage, two of the companies identified themselves as experimental users while the other two were already using the MDE approach in some projects. All companies have identified areas where MDE has added value compared to other development approaches. Value is especially given to:

- Developing domain-specific solutions (in all cases).
- Modeling at different levels of abstraction and from different viewpoints in order to develop more understandable models (in all cases).
- Simulating models in order to assist decision making, evaluating the performance of a system or improving software architecture or design (in three cases).

The benefits lie in better communication among stakeholders, consistency among artifacts, potentially higher productivity if the solution is applied on multiple projects and reduced risk of bad decisions. Compared to the motivations in the published literature (see Section 3.1), we realize that none of the companies taking part in this research have mentioned shorter development time or improved quality of code in single projects as their main motivations for applying MDE. These companies do not look at single projects in an isolated fashion and in the short term, but link benefits to a larger picture of reduced risks and cost-benefit considerations. The participants emphasized that a higher degree of automation and reuse may improve productivity in the long term. There were no special activities on the redesign of development processes. This is required in the future if the MDE approach is to be applied in a structured way.

Regarding intended future use, the current findings do not support the assumption presented in Section 5.2 that current usage motivates increased future usage since the scores and opinions are not related to whether companies are experimental or industrial users. However, companies with experience of applying MDE in real projects perceived MDE as easier to use and more useful (i.e.; companies *A* and *C*). We consider these findings as important. They indicate that companies do not want to miss the advantages of the MDE approach and are receptive to its use, Company *A,* who was an actual user, was most convinced about the benefits of MDE.

For future use, more evidence on the benefits of MDE from research and real projects is needed to convince companies like *B* and *D* to apply the approach in a wider scale, although company *D* has already taken advantage of model-based simulation tools in a few projects.

Companies participating in this research emphasized developing open-source and free solutions as a means to allow free access for experimentation and reduce the cost of adoption. Developing tools based on industry-accepted open platforms, such as Eclipse, has shown benefits such as easier integration. As stated in the interviews, companies raise doubts about the lack of integration between tools and advanced features for managing models of complex systems. In software development, large companies aspire to have a development process with an integrated tool chain that is reusable (with modifications) in multiple projects. The ability to use various MDE tools in different steps of such a process is a strong motivation, while the absence of tools that integrate easily cast doubts on the approach.

8.3. Threats to the Validity of Findings

This is a qualitative study and takes advantage of multiple data sources to discuss the state of practice and acceptance of MDE in the context of four large and complex industrial case studies within a research project. We have tried to address issues important for the quality of this type of study by describing the motivation and context of research, data collection procedures, participants in the research, and discussing the findings and implications including possible bias. We have not produced any statistics based on the questionnaire due to the small sample size, but have summarized the findings per development team and interpreted survey findings together with the findings of interviews and empirical studies performed by the companies.

The companies used MDE at different levels, and thus, the findings reflect diversity between the companies' experiences with MDE. The anonymity of companies allowed us to publish some experiences that would otherwise be kept internal. Threats to validity that apply to our research and implications for interpreting the findings are discussed below.

Firstly, bias can be discussed related to the industrial participants in the research and the interviewer. The participants in this study characterized themselves as moderate to expert developers in MDE. Their experience with MDE varied from 3.5 years to more than 10 years. All interviewees were involved in multiple projects related to MDE and therefore bias towards project solutions in their evaluation of MDE as a methodology is not considered as a threat. However, tools developed in the project and some internal tools used by the industry have been evaluated, which may introduce some bias when discussing opinion regarding tools. We have not compared different tools for solving problems faced by industry, which is a limitation of the study. Interviewer bias is when the interviewer collects data selectively that fit their expectations or theory. The interviewer was not involved in the case studies, and the findings of interviews and the survey were presented to and verified by the involved developers. Informant feedback, or *member checking,* is a technique in qualitative research to improve the viability of the interpretation. Collecting feedback about data and conclusions has been the means to limit the possibility of misinterpretations and any bias in this research.

Bias can also be due to the expectations of companies participating in a research project, especially if the purpose is to advertise a solution or approach. The purpose of this paper is, however, to discuss industrial experiences and not to advocate or criticize a development approach or tool. The study confirmed that adopters were most positive regarding usefulness and ease of use of MDE which is not surprising, while the intentions for future usage did not relate to the current usage. The qualitative techniques are meant to lead to richer information on the subject of the study, but do not allow any claim that the findings are representative of organizations in general. The surveyed developers and architects were among the most advanced MDE users in their companies, and were usually aware of other initiatives in the company, so they can, to some extent, reflect corporate practices in MDE. The small sample size is a threat to the generalization of the questionnaire results to a larger population. However, we do not generalize the findings to the companies and consider the findings as outcomes of case studies performed in these companies. The correct view of generalization is therefore evaluating the findings presented in this paper in the context of reader's experience and how the findings may be valuable in other contexts or in designing future empirical research on MDE.

9. Conclusions and Future Research

Applying a model-driven engineering (MDE) approach is not a straightforward process. It requires developing or customizing several tools, integrating them, convincing managers and developers, training people and changing software processes. The magnitude of impact is often large and there have, so far, been few empirical studies that discuss the factors important for taking a decision on adoption.

This study uses empirical data from four large industrial companies in order to investigate the state of the practice of applying MDE and factors that are considered as important for its adoption. The study complements the findings of previous studies on motivations to use MDE and industrial experiences with MDE, and provides new insights by performing interviews and analyzing four cases together. The goal has been to understand the state of practice, and the opinions of both adopters and those who are still evaluating MDE. We provided concrete feedback from industry on where the approach helps and what should be improved in order to encourage wider acceptance. An extension of the Technology Acceptance Model was selected to investigate the factors important for acceptance, while we provided more evidence from interviews and case studies.

Our findings suggest that the participants in the research found MDE useful for solving some problems of developing complex systems, while the methodology and tools were not generally perceived as easy to use. Both aspects improve, however, with more usage of MDE. Technologically mature tools with features for managing complex models and easy integration with other tools and processes are still missing. Current MDE adopters spend resources on developing internal tools, which illustrates both the shortcomings of tools available in the market and the need for domain-specific solutions and tools that may be tailored to meet users' needs. Additionally, ROI and cost-effectiveness were found to be important factors for the adoption of MDE as emphasized in interviews and research questions. For empirical research on MDE, the findings suggest focusing on MDE promises related to modeling (such as using domain-specific languages, model-based simulation and composition of models) and research on the cost-effectiveness of the approach.

There have been few MDE empirical studies in industry. This comprehensive research that includes data from multiple sources and discusses advantages, problems and factors considered as important for adoption may help those who evaluate MDE, those who develop tools and techniques, as well as empirical researchers to improve their practices. This study has given us an insight in planning empirical research in other projects such as REMICS (REuse and Migration of legacy applications into Interoperable Cloud Services[12]), which also applies MDE techniques. In REMICS, we are looking more into evaluating the impact on the development process and ROI. Future empirical research should cover improving data collection, analyzing success and failure cases and especially evaluating cost-effectiveness of model-driven engineering for different project types and scales.

References

Creswell J.W (2002) Research design – qualitative, quantitative, and mixed method approaches. Sage publications.

Davis F (1989) Perceived usefulness, perceived ease of use, and user acceptance of information technology. MIS Quarterly 13(3):318-339

Davis F, Bagozzi R.P, Warshaw P.R. (1989) User acceptance of computer technology: a comparison of two theoretical models. Management Science 35(8):982-1003

Dybå T, Moe N.B, Mikkelsen E.M (2004) An empirical investigation on factors affecting software development acceptance and utilization of Electronic Process Guides. Proc. Software Metrics, 10th International Symposium (Metrics'04), pp. 220-231

Evans A, Mohagheghi P, Fernández M.A (2009) Experiences of developing a network modeling tool using the Eclipse environment. Proc. 5th European Conference on Model Driven Architecture Foundations and Applications (ECMFA'09), LNCS 5562, pp. 301-312

France R, Rumpe B (2007) Model-driven development of complex software: a research roadmap. International Conference on Software Engineering (ICSE'07), pp. 37-54

Kirstan S, Zimmermann J (2010) Evaluating costs and benefits of model-based development of embedded software systems in the car industry - Results of a qualitative Case Study. Proc. ECMFA 2010 workshop C2M:EEMDD- from Code Centric to Model Centric: Evaluating the Effectiveness of MDD, pp. 18-29

MODELWARE (2006) Deliverable D5.3-1 Industrial ROI, assessment, and feedback- master document, revision 2.2.

Moe N.B, Dybå T (2006) The use of an Electronic Process Guide in a medium-sized software development company. Software Process Improvement Practice 11:21-34

Mohagheghi P, Dehlen V (2008) Where is the proof? A review of experiences from applying MDE in industry. Proc. 4th European Conference on Model Driven Architecture Foundations and Applications (ECMFA'08), LNCS 5095, pp. 432-443

Mohagheghi P, Fernández M.A , Martell J.A, Fritzsche M., Gilani W (2008) MDE adoption in industry: challenges and success criteria. Models in Software Engineering, Workshops and Symposia at MODELS 2008, LNCS 5421, pp. 54-59

Mohagheghi P (2010) An approach for empirical evaluation of model-driven engineering in multiple dimensions. Proc. C2M:EEMDD 2010 workshop at ECMFA 2010- from Code Centric to Model Centric: Evaluating the Effectiveness of MDD, pp. 6-17

Riemenschneider C.K, Hardgrave B.C, Davis F.D (2002) Explaining software developer acceptance of methodologies: a comparison of five theoretical models. IEEE Trans. Software Engineering 28(12):1135-1145

Staron, M (2006) Adopting model driven software development in industry- a case study at two companies. Proc. ACM/IEEE 9th International Conference on Model Driven Engineering Languages and Systems (MoDELS/UML 2006), LNCS, vol. 4199, pp. 57-72

Strauss A, Corbin J (1998) Basics of qualitative research: techniques and procedures for developing grounded theory. Saga Publications

Walderhaug S, Mikalsen M, Benc I, Erlend S (2008) Factors affecting developers' use of MDSD in the healthcare domain: evaluation from the MPOWER project. Proc. From Code Centric to Model Centric Software Engineering: Practices, Implications and ROI. Workshop at European Conference on Model-Driven Architecture

Yin R.K (2002) Case study research – design and methods. Third edition, Sage publications

[12] http://www.remics.eu

Appendix I Semi-Structured Interviews

The following is the list of questions used in the semi-structured interviews:

1. Do you use an MDE approach now?
2. If yes:
   a. What is the main purpose?
   b. Which tools are basically used?
   c. How do you use MDE? Extensive modeling in requirements / analysis / design / testing / system management, generating artifacts from model (M2M or M2T), meta-modeling, separation of concerns in views or CIM/PIM/PSM distinction?
   d. What are the expected benefits? Do you find it useful in your job?
   e. Are you satisfied with the state of the tools regarding ease of use? Problems?
   f. Are you satisfied with the state of the tools regarding tool maturity? Problems?
   g. Are you satisfied with the state of the tools regarding compatibility? Problems?

If not using, why?

3. Do you think there are disadvantages regarding MDE?
4. Which software process do you have? Is that model-based or adapted to MDE?
5. How you describe your software development process now?
6. Which tools are mainly used at the moment?
7. Do you use the following tools and technologies developed in the project? If yes, how? A list of tools and technologies were provided. The scale is: Not used at all, Used occasionally, Used on a regular basis in a few projects, Used on a regular basis in most projects, Used on a regular basis in all projects, Used experimentally in research projects.
8. Are you satisfied with the developed tools regarding ease of use, tool maturity and compatibility?

Appendix II On-line Survey

The questions were distributed randomly in the questionnaire.

Current usage:

1. How would you describe the current usage of the MDE approach in different projects at your department? Scale is: Not used at all, Used on an experimental basis, Used on a regular basis by few projects or people, Used on a regular basis by most projects or people, Used on a regular basis by all projects or people.
2. For how many years have you been using MDE or experimenting with it?
3. How would you describe your expertise in MDE?
4. Which of the following tools have you been using or experimenting with in the project? The list included 23 tools in addition to Eclipse, DSLs and UML profiles. The scale was the same as in Question 1.

Perceived usefulness:

1. I find the MDE approach useful in my job.
2. Using the MDE approach improves my job performance.
3. Using the MDE approach increases my productivity.
4. Using the MDE approach enhances the quality of my job.
5. Using the MDE approach makes it easier to do my job.
6. Overall the advantages of using the MDE approach outweigh the disadvantages.
7. Open-ended question: Based on your experience, what are the main advantages and disadvantages of using the MDE approach?

Perceived ease of use:

1. Learning to use the MDE approach was easy for me.
2. Using the MDE approach does not require a lot of mental effort.
3. I think the MDE approach is clear and understandable.
4. The MDE approach is not cumbersome to use.
5. I find the MDE approach easy to use.

Perceived compatibility:

1. The MDE approach is compatible with the way I develop software.
2. Using the MDE approach is compatible with all aspects of my work.
3. Using the MDE approach fits well with the way I work.
4. The MDE approach is compatible with the way we organize our work.

Perceived maturity of tools:

1. The MDE tools I use are easy to use.

2. The MDE tools I use provide the functionality I need.
3. Using MDE tools improve the way I develop software.
4. The MDE tools I use are compatible with one another and the results can be integrated into one development process.
5. The MDE tools I use are suitable for both small and large projects.
6. The MDE tools I use have acceptable performance.
7. Please assign scores to the tools that you have been using or experimenting with in the project. Scores are: 0- very dissatisfied; 1 – dissatisfied; 2- neither satisfied nor dissatisfied; 3 – satisfied; 4 - very satisfied; 5- don't know.

Subjective norm:

1. Co-workers think I should use MDE.
2. People who influence my work think I should use the MDE approach.
3. People who are important to me think I should use the MDE approach.

Future usage intentions:

2. I intend to increase my use of the MDE approach for work in the future.
3. Given a choice, I would prefer not to use the MDE approach for work in the future.
4. I intend to use the MDE approach in the future for my work.
5. I would like to use the MDE approach in the future for my work.
6. How would you describe your intentions regarding future use of the following tools? The same list as Current usage-Question 4 was provided. The possible responses were: a) Yes, I intend to use or continue in future; b) I will probably use it in future; c) I have not tried it but would like to experiment in future; d) I have not tried and don't think I will do in future; e) I will not use it in future; f) Don't know.

Appendix III Scores Given to Tools and Approaches

The following tables provide scores given to the tools that have been used by more than one company in the project. The scale is: 0- very dissatisfied, 1 – dissatisfied, 2- neither satisfied or dissatisfied, 3 - satisfied, 4 - very satisfied, and don't know. We also asked for the developers' opinions regarding Eclipse as the integration platform in the project, DSLs and UML profiles.

Table 3. Lightweight model repository (prototype developed by Fraunhofer FOKUS)

Company	Functionality	Ease of use	Compatibility	Performance & Reliability	Total impression	Future use intentions
B	3	3	3	4	3	Probably yes
C	3	4	3	3	3	Probably yes

Table 4. Reuseware[11] composition framework for model composition

Company	Functionality	Ease of use	Compatibility	Performance & Reliability	Total impression	Future use intentions
B	4	3.5	4	3.5	3.5	Yes
C	3	2	2	2	2	Probably yes

Table 5. Test derivator and TTCN test generator[13] (prototype developed by Fraunhofer FOKUS)

Company	Functionality	Ease of use	Compatibility	Performance & Reliability	Total impression	Future use intentions
A	3	4	3	3	3	Probably yes
B	3	3	2.5	2.5	3	Yes

Table 6. TraMDE[14] tool for trace definition and analysis

Company	Functionality	Ease of	Compatibility	Performance	Total	Future use

[13] http://www.modelbus.org/modelbus/
[14] http://www.modelbased.net/modelplex/traceability/index.html

		use		& Reliability	impression	intentions
B	2.5	2.5	2.5	2.5	2	Probably yes
D	3	3	3	2	3	Probably yes

Table 7. Model-Driven Performance Engineering tool[15] chain used by company *A* in few projects and by company *C* on an experimental basis

Company	Functionality	Ease of use	Compatibility	Performance & Reliability	Total impression	Future use intentions
A	3.5	3.5	3.5	3	3.5	Probably yes
C	2	3	1.5	2.5	2.5	Probably yes

Table 8. Domain-specific languages

Company	Functionality	Ease of use	Compatibility	Performance & Reliability	Total impression	Future use intentions
A	3	3	2	3	3	Yes
B	3.5	2.5	3.5	3.5	3	Yes
C	3	2	2	2	3	Experimenting
D	3	2	3	3	3	Yes

Table 9. UML profiles

Company	Functionality	Ease of use	Compatibility	Performance & Reliability	Total impression	Future use intentions
A	3	3	1	3	3	Probably yes
B	3	3	2	3	3	Yes
C	3	3	3	3	3	Probably yes[16]
D	3	3	3	3	3	Yes

Table 10. Eclipse as a platform for MDE

Company	Functionality	Ease of use	Compatibility	Performance & Reliability	Total impression	Future use intentions
A	4	4	4	3	4	Yes
B	3.5	4	3.5	3.5	4	Yes
C	4	4	4	4	4	Yes
D	4	4	4	4	4	Yes

[15] http://emftext.org/index.php/MDPE_Contact

[16] They are using UML profiles in real projects. However, UML profiles developed in this project should be finalized before use.