Relevance, benefits, and problems of software modelling and model driven techniques—A survey in the Italian industry

Original
Relevance, benefits, and problems of software modelling and model driven techniques—A survey in the Italian industry / Marco Torchiano; Federico Tomassetti; Filippo Ricca; Alessandro Tiso; Gianna Reggio. - In: THE JOURNAL OF SYSTEMS AND SOFTWARE. - ISSN 0164-1212. - STAMPA. - 86:8(2013), pp. 2110-2126. [10.1016/j.jss.2013.03.084]

Publisher:
Elsevier

Published
DOI:10.1016/j.jss.2013.03.084

Relevance, Benefits, and Problems of Software Modelling and Model Driven Techniques - A Survey in the Italian Industry

Marco Torchiano[a,*], Federico Tomassetti[a], Filippo Ricca[b], Alessandro Tiso[b], Gianna Reggio[b]

[a]*Dip. Automatica e Informatica, Politecnico di Torino, Italy*
[b]*DIBRIS, Università di Genova, Italy*

Abstract

Context: Claimed benefits of software modelling and Model Driven techniques are improvements in productivity, portability, maintainability and interoperability. However, little effort has been devoted at collecting evidence to evaluate their actual relevance, benefits and usage complications.

Goal: The main goals of this paper are: (1) assess the diffusion and relevance of software modelling and MD techniques in the Italian industry, (2) understand the expected and achieved benefits, and (3) identify which problems limit/prevent their diffusion.

Method: We conducted an exploratory personal opinion survey with a sample of 155 Italian software professionals by means of a web-based questionnaire on-line from February to April 2011.

Results: Software modelling and MD techniques are very relevant in the Italian industry. The adoption of simple modelling brings common benefits (better design support, documentation improvement, better maintenance, and higher software quality), while MD techniques make it easier to achieve: improved standardization, higher productivity, and platform independence. We identified problems, some hindering adoption (too much effort required and limited usefulness) others preventing it (lack of competencies and supporting tools).

Conclusions: The relevance represents an important objective motivation for researchers in this area. The relationship between techniques and attainable benefits represents an instrument for practitioners planning the adoption of such techniques. In addition the findings may provide hints for companies and universities.

Keywords:
Model Driven techniques, Software modelling, Personal Opinion Survey, Relevance, Benefits, Problems.

---

1. Introduction

Usually, model-based techniques use models to describe the architecture and design of a system and/or the behaviour of software artefacts generally through a raise in the level of abstraction [11]. Models can also be used for other purposes, e.g., to describe business workflows or development processes.

They can be used in different phases of the development process as communication artefacts, as points of references against which subsequent implementations are verified, or as the basis for further development. In the latter case, they may become the key elements in the process, from which other artefacts (most notably code) are generated.

The term "model" is very general; it is difficult to provide a comprehensive yet precise definition of a model. For us, a model is an artefact realized with the goal to capture and transfer information in a form as pure as possible limiting the syntax noise. Examples of models include UML design models [35], process models defined through BPMN [34], Web application models defined through WebML [5] as well as textual Domain Specific Language (DSL) models [10].

Given that models are so heterogeneous and the processes involving them so varied, it is actually difficult to define the exact boundaries for modelling. Moreover, models can be used practically in different ways and with different levels of maturity [40]. While in theory we have a set of best practices, in practice we only

---

[*] Correspondig author:
Marco Torchiano
Dip. Automatica e Informatica, Politecnico di Torino,
C.so Duca degli Abruzzi 24, 10129 Torino, Italy
Phone: +39 011 564 7088, Fax: +39 011 564 7099
Email: `marco.torchiano@polito.it`

found mixed, half baked, and personalized solutions. As a consequence, the different uses of modelling are complex and difficult to classify precisely.

When models constitute a crucial and operational ingredient of software development then the process is called model driven. There are different model driven techniques: Model Driven Engineering (MDE) [37], Model Driven Development (MDD) [28] and Model Driven Architecture (MDA) [22]. MDE is the broadest one; it is a software methodology that focuses on creating and exploiting models, which are typically given as abstract representations using a modelling language (e.g., UML but also BPMN or a home-grown DSL [10]). Instead, MDD is a development approach that uses models as the main artefacts of the development process. In MDD, models at higher-level of abstraction are (progressively) transformed into models at lower-level of abstraction until the models are executable using either code generation or model interpretation. In this latter case, executable models are directly executed/interpreted by means of specific environments [27]. The main difference between MDE and MDD is that MDE goes beyond the pure development activities and encompasses also other software engineering tasks (e.g., models evolution and reverse engineering of legacy systems). Finally, MDA is a registered trademark of Object Management Group (OMG) recommending the usage of OMG standards (e.g., UML). MDA is more specific than MDD and defines the system functionality using the notions of platform-independent model (PIM) and platform-specific model (PSM). We are interested in no specific model driven technique, therefore, in the following, taking inspiration from [43], we will address all these related technologies collectively with the abbreviation MD*.

Although a lot of work has been done in the MD* context and, a wide number of studies have been reported in literature (e.g., [17, 18, 31]), and many commercial or free MD* tools are also available (e.g., UniMod [13], WebRatio [1] and BridgePoint[1]), there is a lack of evidence about the relevance of MD* in industry, and we need indication whether (or not) MD* satisfies today's industry needs [29].

For those reasons, two Italian universities, Politecnico di Torino and Università di Genova, started a common project concerning software modelling and MD*. The first step of the project aimed at achieving an accurate picture of the state-of-the-practice of modelling and MD* in the Italian industry by means of a survey of the Italian ICT industry. We opted for a personal opinion survey[2] [12, 21] performed through the Internet, because this is generally the most cost effective interview method [44] even if it presents well-known limitations/problems [39].

This work is based on the same data previously analyzed in [40, 42] but extends our earlier work in different ways and focuses on different aspects, not yet explored, of the modelling/MD* phenomenon. In particular, in the short paper [42] we gave some preliminary findings about our survey, considering which processes, languages and tools are used in the modelling/MD* context. On the contrary, the main goal of [40] was investigating the level of maturity in the usage of software models and MD* in the Italian industry. Instead, here, we focus our interest on three key aspects: (1) relevance of modelling/MD* measured using their real diffusion, (2) benefits of using modelling/MD* and, (3) problems hindering or preventing modelling/MD*.

The evidence we collected about modelling and MD*, by means of this survey, holds a value "per se" as new assets in the software engineering body of knowledge. In addition, it brings important implications in the practice of both software development and education/training. We think that, on the basis of the results of this broad survey, more specific studies could be conducted to confirm and clarify the most controversial or difficult understandable findings.

The paper is structured as follows. Section 2 presents the relevant aspects of the conducted survey such as: goals, research questions, questionnaire design, sample identification, survey preparation/execution and analysis methodology. Section 3 presents the findings about relevance, benefits and problems of modelling and MD*. Section 4 discusses the obtained results and Section 5 examines the unavoidable threats to validity. Section 6 is about related works and, finally, Section 7 concludes the paper and presents ideas for future work.

2. Study definition

The instrument we selected to take a snapshot of the state of the practice concerning industrial MD* adoption is that of a survey [12]. In the design phase of the survey we drew inspiration from previous surveys (i.e., [20, 23, 41]) and we followed as much as possible the guidelines provided in [21].

---

[1] http://www.mentor.com/products/sm/model_development/bridgepoint/

[2] The purpose of a personal opinion survey is to produce statistics, that is, quantitative or numerical descriptions of some aspects of the study population.

The survey has been conducted through the following six steps [21]: (1) the objectives (or goals) selection, (2) goals' transformation into research questions (section 2.1), (3) questionnaire design, (4) sampling and evaluation of the questionnaire by means of pilot executions, (5) survey execution and, (6) analysis of results and packaging.

We conceived and designed the survey with the goals of understanding:

G1 the actual relevance of software modelling and MD* in the Italian industry,

G2 the way software modelling and MD* are applied (i.e., which processes, languages and tools are used), and

G3 the motivations either leading to the adoption (expected benefits) or preventing it (experienced or perceived problems).

The above three goals cover quite a wide spectrum; in this work, we mainly focus on the goals G1 and G3. As far as goal G2 is concerned, some preliminary findings can be found in [42] and a brief summary is provided in Section 3.2.

2.1. Research Questions

Within the scope of this work, we aim at addressing three main research questions (with relative sub-questions), one (RQ1) is related to goal G1 and the other two (RQ2 and RQ3) are related to goal G3:

RQ1: What is the diffusion and relevance of modelling and MD* in the Italian industry?

Finding out the proportion of IT professionals actually adopting modelling and MD* should allow us to understand how important such development techniques are. Knowing whether they represent niche, commonly used, or mainstream methodologies has a dramatic impact on the conclusions we can draw.

RQ1.1: What is the adoption ratio of individual MD* techniques?

Since MD* is typically applied with diverse blends of basic techniques, we want to examine which techniques are used (e.g., code generation or model execution) and which are the most relevant.

RQ1.2: What is the diffusion by company size category?

Some authors (e.g., Selic [38]) believe MD* is mostly intended for large projects carried on by large companies; is it true? An empirical validation of such a claim requires breaking down the prevalence figures by company size.

RQ2: What are the benefits of using modelling and MD*?

From a technology transfer perspective it is important to motivate the adoption of a new technique by presenting the benefits that it could bring. Therefore, we are interested in understanding which specific technique (e.g., code generation) increases the likelihood of a given benefit.

RQ2.1: Do individual MD* techniques affect the achievement ratio of specific benefits?

RQ2.1: Which benefits are most common in each company size category?

RQ3: What issues hinder/prevent the adoption of modelling and MD*?

The reasons that hinder/prevent the adoption of modelling and MD* are as important as the potential benefits. In some cases these motivations may exclude such techniques from the whole company, in other cases the preclusion may be on a per-project basis.

2.2. Population and sampling strategy

The first step to conduct a survey consists in defining a target population. In our study the target population is formed by software development teams or business units. To get information about the target population we defined a framing population consisting of Italian software professionals – i.e., project managers, software architects, developers – whom we asked to answer our questions (about the target items).

To sample the population we applied both probabilistic (random sampling) and non-probabilistic (convenience sampling) methods [21]. More in detail, the sampling was performed in five ways: (i) randomly selecting contacts from the the Commerce Chamber database, (ii) selecting contacts from the industrial contact networks of the two research units involved (Torino and Genova), (iii) sending invitation messages to mailing lists concerning programming and software engineering, (iv) publishing an advertisement in an on-line magazine for developers (programmazione.it) and, and (v) placing and advertisement on the web portal of a large Italian developers' conference (CodeMotion 2011).

We decided to collect data through an on-line questionnaire created by means of the LimeSurvey survey tool[3]. Web-based questionnaires, compared to paper-based questionnaires or email-based questionnaires, allow an easier data entry from the respondent perspective, a simpler data collection from the researcher perspective and are less error prone [36]. In general, it has been observed that Web-based questionnaires guarantee high return rates [20].

2.3. Survey Preparation and Execution

The procedure followed to prepare, administer, and collect the questionnaire data is made up of the following five main steps.

Preparation and design of the questionnaire. We first prepared a preliminary version of the questionnaire. Then, we conducted three different pilots with software professionals to identify any problems with the questionnaire itself [21], before putting on-line the final version. According to the feedback obtained, we made a few changes to the questionnaire to improve the validity of the instrument.

On-line deployment. Once finalized, the questionnaire was uploaded to the LimeSurvey server to enable the automatic collection of data.

Invitation to participate. Organizations were sampled as detailed above (Section 2.2). For the contacts that we selected directly, once the contact persons were identified, we invited them, via email, to register with the survey server and to complete the on-line questionnaire. We also broadcast invitation on selected mailing lists and on-line magazines/conferences including in the message a link ("click here to take the survey") to a registration form where the participants could register themselves and fill in the questionnaire. The questionnaire was introduced by a brief description page summarizing goals and motivation of this study and it was accompanied by a cover letter briefly introducing our research project. In the cover letter we tried to summarize: the purpose of the study, the relevance to the participants and, why each individual participation was important [21]. Great care was taken to ensure that ethical requirements and privacy rules imposed by the Italian regulations were met[4]. For example, they require confidentiality but give the possibility of publishing the results in aggregated form. We decided to avoid any form of material incentives for participation. However, to motivate professionals, we promised to provide a report containing the analyses and the obtained results to all participants.

Monitoring. During the data collection phase, we monitored the progress of the questionnaire submission. This allowed us to send selective reminders to contacts who did not respond or did not completed the questionnaire yet. Some people reported some difficulties about the questions, either due of internal policies of the company or because involved in very different projects with different companies at the same time; they asked us some clarifications about the questions.

Data analysis and Packaging. After data collection was concluded, we performed analyses as described in section 2.5 and we packaged instruments, data, and results in a replication package.

2.4. Questionnaire Design

The questionnaire was developed to directly address the goals of the study. To harvest as many responses as possible, we designed the questionnaire to limit as much as possible the time necessary to complete it, having in mind that long questionnaires get less response than short ones [44][5].

The questionnaire contains a total of thirty-one items, both open and multiple-choice expressed in Italian language. However, the actual number of items administered to any individual respondent depends on her adoption of MD* and modelling (e.g., respondents not adopting modelling in their software process were required to answer eight questions only).

The structure and possible paths through the questionnaire are described in Figure 1. The questionnaire consists of four sections; each session is identified by a three characters identifier (Sub, Dev, Mod, and Lan) and is described below. The individual items are named using the section identifier followed by a two digits progressive number (e.g., Sub04 is the fourth item in the questionnaire and appears in the Sub section).

Sub (subject's demographics): this is the first section and is administered to all respondents; its goal is to characterize the respondents and their organization. In particular, it collects: business domain, organization size, respondent's group/business unit size and experience of unit members. For example, freelance or individual companies are asked only

---

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

[4] privacy Italian law: "D.lgs. n.196/2003".

[5] It turned out the actual time for completing the questionnaire was on average less than six minutes.

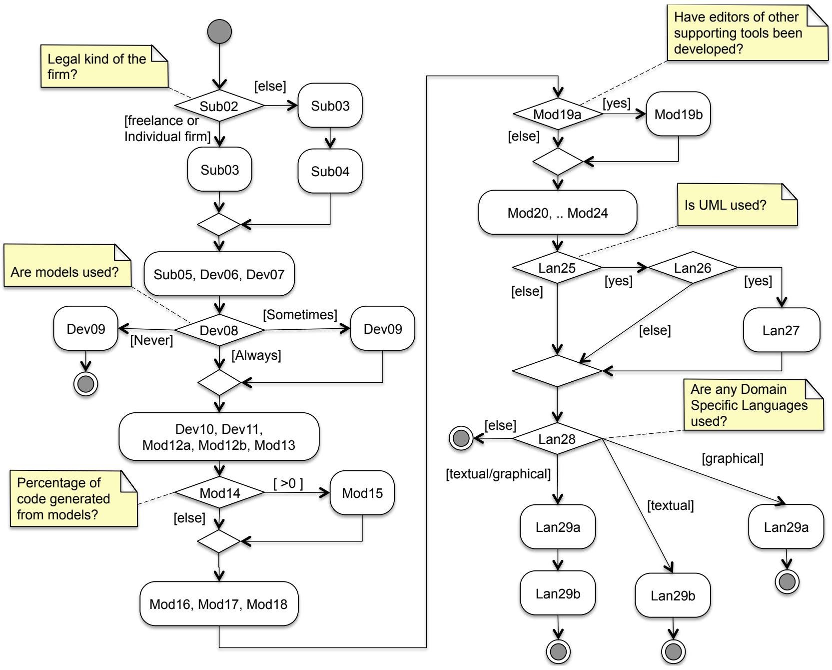


Figure 1: Questionnaire structure.

three distinct questions of this kind (Sub02, Sub03 and Sub05), as we may observe from Figure 1.

Dev (development process): the second section collects information concerning the kind of projects conducted, their average duration, whether the respondent uses models, and the expected and achieved benefits. The respondents that do not use modelling systematically (i.e., answering *never* or *sometimes* at question Dev08) are asked about the problems preventing or limiting to use models (see question Dev09 in Table 1 and Figure 1). Respondents never using modelling terminate the questionnaire with this section.

Mod (modelling details): the items in this section are administered to respondents that use modelling at least sometimes. The section collects information concerning the adoption of processes, techniques and tools.

Lan (languages and notations): contains items measuring use of UML, UML profiles and domain specific languages (DSLs).

Table 1 reports the subset of the items in the questionnaire that are relevant to the research questions considered in this paper. For each question the table reports the identifier, the question (translated into English from Italian), the type of measure, and the RQ it is relevant to. The complete questionnaire (in Italian) is available for downloading on the web[6].

A few items in the questionnaire are particularly important for the purpose of this paper and deserve more attention.

In section Dev, the most important item is Dev08 that corresponds to the following question: *"Are models used for software development in your organization?" By model we mean both diagrams and text artefacts created using either general purpose modelling languages (e.g., UML) or Domain Specific Languages (DSLs)*[7]. Dev08 is a closed question that allows three valid answers: *always*, *sometimes* and *never*. Depending on the Dev08 answer, three distinct paths were followed (see questionnaire structure in Figure 1):

- respondents that *always* use modelling were asked about the benefits (Dev11),
- those using modelling only *sometimes* were asked about the benefits (Dev11) and also about the problems preventing the use of modelling (Dev09), and

---

[6] *http://softeng.polito.it/tomassetti/MDQuestionnaire.pdf*

[7] This was clarified in the questionnaire given to the participants.

ID	Question	Type	RQs
Sub02	What legal entity/kind does your company fit?	Nominal	RQ1,RQ3
	Valid answers: Freelance/individual firm; Firm/company; Public institution; Other		
Sub03	What is the main business activity of your company?	Nominal	RQ1,RQ3
	Valid answers: [Manufacturing; IT; Public Administration; Service Provider; Transport; Telecom; Other]		
Sub04	How many persons does your company count, including part-time, full-time staff and consultants?	Ordinal	RQ1,RQ3
Sub05	Provide the experience (in years) of the business unit's members	Interval	RQ1,RQ3
Dev08	Are models used for software development in your organization? (for model we mean both diagrams, e.g., UML, and text according to any DSL)	Nominal	RQ1,RQ3
	Valid answers: Always; Sometimes; Never		
Dev09	What are the problems preventing modelling and MD*?	Nominal	RQ3
Dev11	What are the benefits expected and verified from using modelling and MD*?	Nominal	RQ2
Mod14	What is the percentage of code generated from models?	Interval	RQ1
Mod14*	Is code generated? *Derived measure:* $Mod14* = Mod14 > 0$	Yes/No	RQ1
Mod16	Are models executed (interpreted) at run-time?	Yes/No	RQ1
Mod18	Are transformation languages (e.g., ATL) used?	Yes/No	RQ1
Mod19a	Have modelling support tools and editors been developed?	Yes/No	RQ2
MD_USAGE	Is any MD* technique used?	Yes/No	RQ1
	Derived measure: $MD\_USAGE = Mod14 * \vee Mod16 \vee Mod18$		
Lan25	Is UML used?	Yes/No	RQ2
Lan26	Are UML profiles used?	Yes/No	RQ2
Lan28	Are Domain Specific Languages used?	Nominal	RQ2
	Valid answers: No; Yes, textual; Yes, graphical; Yes, both graphical and textual		

Table 1: Subset of relevant questionnaire items (translated from Italian to English). Questions are condensed in respect to the administered survey.

- respondents *never* using modelling were asked only about the potential problems (Dev09).

Given their importance, we evaluated accurately the possible answers presented for questions Dev09 and Dev11 in the design phase of the questionnaire. Moreover, we fine-tuned them based on the outcomes of the pilots. The values we chose are reported in Figure 2.

Since we were not sure about the range of possible problems preventing the usage of models and MD*, we designed item Dev09 as a multiple answers question with a set of predefined options plus an additional open option.

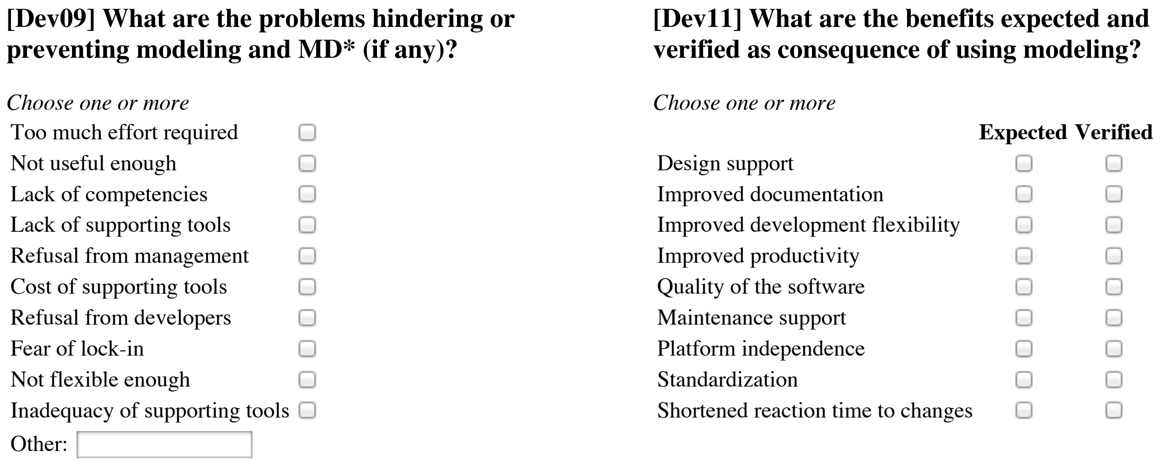
Conversely, after the feedback from the pilots, we were confident we identified all the significant benefits, therefore item Dev11 was designed as a closed answer with multiple choices. The question Dev11 asks which benefits among the ones presented were expected by the respondents and which ones were actually verified. For each benefit the respondent had the possibility to mark separately if the benefit was expected or verified. In this way four combinations are possible for each single benefit: it could have been expected and verified (positive confirm), expected and not verified (negative surprise), not expected and verified (positive surprise) or not expected and not verified (negative confirm).

In the questionnaire there are three items concerning MD*-specific practices: code generation (Mod14), model interpretation (Mod16), and model transformations (Mod18). We plan to handle the three practice categories in an homogeneous way; since Mod14 measures the percentage of code generated from models, while the other two (Mod16 and Mod18) just measure the adoption as a boolean value, we introduced a derived variable (Mod14*) with boolean type whose value is true when some code is generated and false otherwise, i.e., $Mod14* = Mod14 > 0$.

We assume that using at least one of the three techniques means adopting MD*, therefore, we defined a derived item (MOD_USAGE see Table 1) that is defined on the basis of the three technique oriented items, i.e., $MD\_USAGE = Mod14* \vee Mod16 \vee Mod18$.

2.5. Analysis methodology

We address the three research questions delineated in Section 2.1 by means of descriptive statistics and where



[Dev09] What are the problems hindering or preventing modeling and MD* (if any)?

Choose one or more

Too much effort required	☐
Not useful enough	☐
Lack of competencies	☐
Lack of supporting tools	☐
Refusal from management	☐
Cost of supporting tools	☐
Refusal from developers	☐
Fear of lock-in	☐
Not flexible enough	☐
Inadequacy of supporting tools	☐
Other:	

[Dev11] What are the benefits expected and verified as consequence of using modeling?

Choose one or more

	Expected	**Verified**
Design support	☐	☐
Improved documentation	☐	☐
Improved development flexibility	☐	☐
Improved productivity	☐	☐
Quality of the software	☐	☐
Maintenance support	☐	☐
Platform independence	☐	☐
Standardization	☐	☐
Shortened reaction time to changes	☐	☐

Figure 2: Options presented in questions Dev09 and Dev11.

applicable by statistical hypothesis testing, and show our findings by means of graphs.

Statistical correlation between a factor (e.g., using UML profiles) and the benefit achievement ratio (the proportion of respondents who achieved each specific benefit) are verified by means of standard tests. In particular, we opted for non-parametric tests because of the nature of the variables, which are measured on nominal and ordinal scales. Specifically, according to the recommendations given in [2, 32], we used:

- the Fisher exact test for the correlation among two dichotomous variables,
- the χ^2 test for the correlation among categorical variable variables,
- the Mann-Whitney tests for testing the difference between two groups (we can interpret a significant MW test as showing a difference in medians), and
- the Kruskal-Wallis test for three or more groups.

The decision whether rejecting or not the null hypotheses verified by statistical tests is taken considering a level of significance of 95%. This means that when we draw our conclusions, we accept a probability ($\alpha = 5\%$) of incurring in a type I error (i.e., rejecting a null hypothesis when it is true).

We present now the method used to specifically address each research question.

2.5.1. *RQ1: diffusion and relevance*

We answer RQ1 by looking at one dependent variable: the answers to the item about model usage (Dev08). In particular, we focus on the frequency of the three valid answers (*Never*, *Sometimes*, and *Always*). The analysis of both this RQ and the following ones, will focus on the “modellers”, with this term hereinafter we refer to the respondents who use modelling at least sometimes: that is those who answered *Sometimes* or *Always* to item Dev08.

In agreement with the definition of relevance provided in [16], we can state that the study of a specific software development technology is relevant to software engineering if it increases the likelihood of improving software development practices. Such a perspective involves both technical aspects, which are out of the scope of our investigation, and process aspects which are in part addressed in our investigation. With a little bit of simplification we assume here that the main process factor for evaluating the diffusion of a technology is the proportion of developers that use it. In the context of this study, we assume a proportion larger than 50% implies a high relevance, larger than 25% a normal relevance, larger than 10% a limited relevance, and below 10% irrelevance. We defined the above thresholds in an arbitrary way using just common sense.

We compute the relevance level (high, normal, limited, irrelevant) for modelling by comparing the above thresholds to the confidence intervals for the proportions. In particular, we compute the 95% confidence interval (CI) by means of the proportions test [2]. Then, to be conservative as much as possible, we assign the level corresponding the highest threshold that is smaller than the lower limit of the confidence interval. For instance if the 95% CI of the diffusion of a technology is

[35% , 65%] we assign the “normal relevance” category since the corresponding threshold (25%) is the highest one smaller than the lower limit (35%).

Then, we analyse the relevance and diffusion indicators with respect to the company size (Sub04). In particular, we check for the existence of a correlation between the adoption ratio of modelling and the company size categories. To this end, we build a 2×5 contingency table of modelling adoption (sometimes or always) vsṅo modelling on one side and company size categories on the other side (we considered five company size categories, see Table 2). We apply the χ^2 test to reject the null hypothesis that no correlation exists. In addition we evaluate the relevance for each company size category

Moreover, we focus on the adoption of the MD* specific practices: code generation (Mod14*), model interpretation (Mod16), and model transformations (Mod18). In particular, we assume that respondents using at least one of the above cited techniques is adopting MD* [8]

For both the general MD* adoption and each specific technique we perform the same analyses as above: first we classify the relevance and then we analyze it in relation to company size.

2.5.2. *RQ2: benefits*

Question RQ2 is addressed by analysing the answers to a composite item (Dev11) which listed several potential benefits and asked which were expected and which were achieved (see Figure 2). For this question we focus on the actually achieved advantages. In particular we adopt as metric the benefit achievement ratio: the proportion of respondents who achieved each specific benefit.

As a preliminary step, we investigate the cross correlation among the different benefits, since the adoption is a dichotomous variable, we selected the Pearson’s ϕ as a strength of correlation measure. The statistical significance of the correlation is verified by means of the Fisher exact test.

We first report the benefit achievement ratio among all modellers, with the intention of classifying the benefits in terms of their likelihood. In particular we assume that above a 50% frequency a benefit can be considered as *Very Likely*, above 25% as *Likely*, above 10% as simply *Probable*, and below that threshold as *Unlikely*. As for the relevance categories of RQ1, we compute the 95% confidence interval of the proportion using a proportion test and compare the lower limit with the above thresholds.

Then, we report the proportion of respondents who achieved each specific benefit making a distinction between adopters of simple modelling and adopters of MD*. We investigate whether a significant difference in benefit achievement ratio between the two groups exists and for any such case we will perform a further classification of the relative likelihood. For the purpose of identifying the difference we observe the odds ratio[9] of benefits achievement for the two groups and test the significance by means of the Fisher exact test.

After that, we focus on the three key MD* techniques (i.e., code generation, model transformation and model interpretation). In particular, we check whether the adoption of a single technique induces a significant difference in terms of benefit achievement ratio. As for the previous step, we focus on odds ratios and we use the Fisher exact test to identify significant differences.

Then, we check for the existence of a correlation between the achievement ratio of each individual benefit and the company size categories. Similarly to the procedure adopted for RQ1, for each benefit, we build a 2×5 contingency table of achievement vs. not achievement of a benefit on one side and company size categories on the other side. We apply the χ^2 test to reject the null hypothesis that no correlation exists.

Eventually, as a last step concerning RQ2, we observe the relationship between benefit achievement and other factors, namely toolsmithing, adoption of UML, UML profiles, and DSLs. In particular we observe how the odds of achieving a benefit change when each individual technique is employed. The Fisher exact test is used to test the null hypothesis that no significance difference exists.

2.5.3. *RQ3: problems*

To answer RQ3, we consider the composite item Dev09 which reports for each participant a list of problems (s)he found prevented her/him from adopt modelling (see Figure 2). The respondents to this item include those who never use modelling and those who use

---

[8] With respect to our previous paper [42], here we use a different definition for MD* adopter.

[9] The odds ratio is a measure of effect size that can be used for dichotomous categorical data. An odds indicates how likely it is that an event will occur as opposed to it not occurring. Odds ratio is defined as the ratio of the odds of an event occurring in one group to the odds of it occurring in another group. An odds ratio of 1 indicates that the condition or event under study is equally likely in both groups. An odds ratio greater than 1 indicates that the condition or event is more likely in the first group. Finally, an odds ratio less than 1 indicates that the condition or event is less likely in the first group.

Interval	Group	Frequency
1	Individual	24
2-10	Micro	25
11-50	Small	26
51-250	Medium	26
251+	Large	54

Table 2: Frequency of respondents for different company sizes.

Code Generation	○	●	○	○	●	●	○	●
Model Interpretation	○	○	●	○	●	○	●	●
Model Transformation	○	○	○	●	○	●	●	●
Freq	55 52%	30 28%	3 3%	0 0%	6 6%	3 3%	1 1%	7 7%

Table 3: Combined diffusion of MD* techniques (●: technique used, ○: technique not used).

it sometimes: for the former the problems prevent altogether modelling, while for the latter the problems just curb it.

We adopt a similar approach as for RQ2, therefore we first verify cross correlation among problems, then we report the problem occurrence ratio among modellers, with the intention of classifying them in terms of relevance. For this purpose we adopt the same criteria used in RQ1.

In analogy with the procedure adopted for RQ2, we eventually divide the respondents into two groups – those who never adopted modelling and those who used it sometimes – and we analyse the problems considering their relationship with company size.

3. Findings

First of all we present the characteristics of the population sample, then we present the results concerning the three research questions.

3.1. The sample

The survey was put on-line from the 1st of February 2011 until the 15th of April 2011 (two and a half months).

In total, we collected 155 complete responses to our survey, thus the context of our survey consists of a sample of 155 Italian software professionals. Due to the sampling methods used, it is not possible to estimate how many people have been reached by our invitation messages and/or advertisements; as a consequence it is not possible to compute the response rate. This limitation appears to be a common problem for large scale online surveys (see, e.g., [24]).

As far as the type of company is concerned, most respondents in our sample – 122 out of 155 (78.71%) – work in commercial companies; there are also 24 independent professionals (15.48%), six from public organizations (3.87%) and three from other organizations. Concerning the domain, the most represented sector is obviously *IT* (67%) followed by *Services* (9%), *Telecommunications* (6%) and *Manufacturing* (4%); the remaining sectors all together account for circa 14% of the sample.

The respondents to our survey belong to companies of different size; the detailed distribution for each category class is reported in Table 2. We adopted the headcount classes defined in the European Union recommendation 2003/361/EC[10].

It is important to emphasize that in our sample the correspondence between respondents and companies is not strictly one-to-one. In some cases we had more than one response from employees of the same company: in such cases we verified that they worked in distinct business units. This is obvious when consider that the target population consists of development teams and companies, especially the large ones, typically host several business units and work groups, each possibly working in different settings.

3.2. RQ1: relevance and diffusion

We first consider relevance of modelling in general, then we summarize languages and notations used by our sample [42]. Finally, we examine the diffusion of each specific MD* techniques.

3.2.1. Relevance of modelling in general

Among the 155 complete questionnaires, we recorded 20 respondents (13%) always using modeling, 85 (55%) using it sometimes, and 50 (32%) who never use it. The first two groups represent what we called ”modellers”. These figures are reported in the rightmost bar of Figure 3. The circle represents the estimate proportion of modellers in the sample (68%) and the whiskers represents the 95% confidence interval (CI) of the same proportion.

We can classify the relevance of modelling by comparing such CI with the thresholds defined in Section 2.5.1 (50%, 25%, and 10%), which are represented by the horizontal dashed lines. The 95% CI of the proportion of modellers is [60%, 75%], the lower bound is larger than the 50% threshold, therefore modeling can be classified as a *highly relevant* technology. We can

---

[10] http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=OJ:L:2003:124:0036:0041:EN:PDF. Note that in our previous work [42] we used a slightly different division.

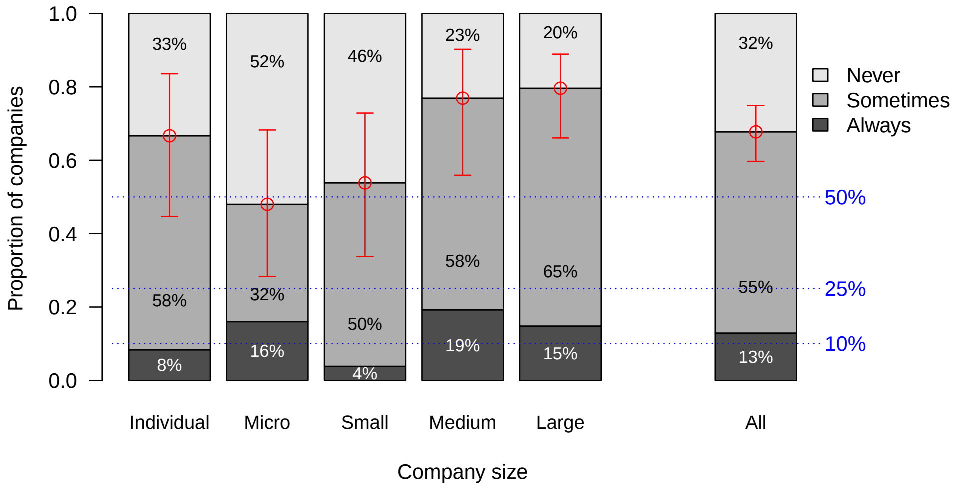


Figure 3: Proportion of modeling usage per company size.

get to the same classification by graphically comparing the lower end of the CI with the reference dashed lines: the whiskers lie completely above the 50% reference.

The proportion of developers adopting modelling varies significantly (χ^2 test p = 0.02) with the size of company as we can appreciate also in Figure 3. According to the 95% CI we can still classify modeling as highly relevant for medium and large companies, while it can be considered as simply relevant for the other companies.

We observe that, with the exception of ”individual” companies, the use of modelling (i.e., always + sometimes) is positively correlated with the size of companies (i.e., it is more frequent in large companies). Individual companies represent an exception to that trend since they are closer to large companies' levels. Concerning the systematic use of modelling (i.e., always), we observe a similar diffusion at micro, medium, and large companies; while apparently there are few small-sized companies and individuals that systematically adopt modelling practices.

3.2.2. Languages and notations

Among the 105 modellers, 80 of them (76%) adopt UML as modelling language (Lan25). Among them (Lan26), 11% use also UML profiles, 51% do no use them, and the remaining 38% state to not know if them are used in their organization.

In our sample, only 21% of modellers appear interested in Domain Specific Languages (Lan28). Among them 50% use a purely textual notation, 23% a purely graphical one, and 27% a mix of textual and graphical notations.

3.2.3. Relevance of MD specific techniques*

Among the 105 respondents that use modelling, 50 of them (48%) adopt at least one of the three key MD* techniques. The 95% CI of the proportion of developers among using MD* is [25% , 40%] of all developers, therefore we can classify MD* as a *relevant* development technology.

The relative frequency of adoption of the MD* specific practices among modellers is depicted in Figure 4. Overall, we observe that, in our sample, code generation is in use by 44% of the 105 modellers, model interpretation by 16%, and model transformation by 10% (not in exclusive way). The 95% CI of the frequency of the use of the individual techniques by all developers are [34% , 54%] for code generation, [10% , 25%] for model interpretation, and [6% , 18%] for model transformation. We can compare them to the relevance thresholds defined in section 2.5.1. Therefore code generation can be classified as a *relevant* technology, model interpretation as a technology with *limited relevance*, while the diffusion of model transformation is not enough to consider it relevant for the practitioners.

If we narrow down our scope to MD* adopters, only, 46 out of 50 MD* adopters (92%) use code generation, 34% use model interpretation, and 20% use model transformation.

When we focus on the adoption of (any) MD* specific techniques as a function of the company size we observe a sort of bimodal distribution. A very large pro-

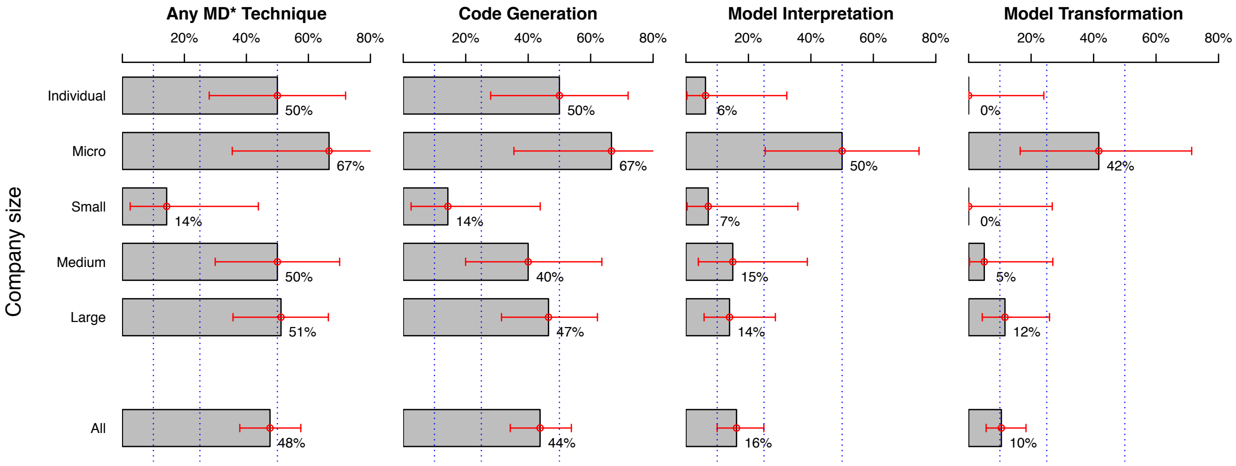


Figure 4: Diffusion of MD* techniques among modellers per company size.

		Achievement ratio			MD* vs. Basic modeling		Code generation		Model interpretation		Model transformation	
Benefit	Freq.	Estimate	95% CI	Likelihood	OR	p	OR	p	OR	p	OR	p
Design	71	68%	58% .. 76%	Very likely	1.23	0.68	1.0	1.00	2.5	0.26	1.3	1.00
Documentation	65	62%	52% .. 71%	Very likely	0.62	0.31	0.6	0.22	0.9	0.79	0.5	0.33
Maintenance	43	41%	32% .. 51%	Likely	1.49	0.33	1.2	0.69	1.8	0.29	2.8	0.12
Quality	42	40%	31% .. 50%	Likely	2.22	0.07	1.8	0.16	1.9	0.28	2.9	0.11
Standardization	39	37%	28% .. 47%	Likely	2.44	**0.04**	2.2	0.07	1.6	0.42	1.5	0.53
Flexibility	24	23%	15% .. 32%	Possible	2.17	0.11	1.4	0.49	**3.9**	**0.02**	2.1	0.27
Productivity	23	22%	15% .. 31%	Possible	5.53	< 0.01	**3.9**	**0.01**	**4.2**	**0.01**	**8.3**	**<0.01**
Reactivity to changes	20	19%	12% .. 28%	Possible	1.84	0.32	1.1	1.00	**4.0**	**0.02**	2.7	0.21
Platform independence	15	14%	8% .. 23%	Unlikely	5.39	**0.01**	3.0	0.09	**4.7**	**0.02**	**4.2**	**0.05**

Table 4: Benefits achieved by modelling users (OR=Odds Ratio, p=Fisher test p-value).

portion (67%) of the micro companies adopt MD* techniques, medium and large companies have an adoption ratio of circa 50%. This trend is completed by a sudden drop in adoption for small companies where just 14% adopt such techniques (see Figure 4 leftmost plot).

Considering each technique alone, we observe a similar shape for the distribution but with some notable differences. The adoption of code generation is very similar to the adoption of MD* techniques: this is obvious since it is by far the most widespread among the three techniques. As far as model interpretation and model transformation are concerned, micro companies are the only significant adopter, large companies adopt them very marginally and still small sized companies exhibit little or no interest in them.

The above picture can be drawn considering the adoption of the techniques as separate; in practice specific techniques are adopted both individually ($^2/_3$ of the cases) and in combination with each other ($^1/_3$ of the cases). Table 3 reports the relative frequency of the different combination of techniques (indicated in the table with black circles) that were found in use among the respondents. We observe that the most common toolbox consists of code generation alone (28% of modelers), the next most frequent sets are the combination all the three techniques (7%) and the use of code generation together with code model interpretation (6%). The other options are adopted by a few respondents. Notably, model transformation techniques are never used alone but only together with other techniques (that is pretty obvious).

	Toolsmithing		UML		UML Profile		DSL	
Benefit	OR	p.value	OR	p.value	OR	p.value	OR	p.value
Design	1.2	1.00	2.4	0.09	4.1	0.25	1.3	0.62
Documentation	0.5	0.18	2.6	0.06	1.1	1.00	0.9	0.81
Maintenance	2.4	0.11	2.1	0.16	1.9	0.46	1.0	1.00
Quality	1.9	0.28	1.2	0.82	1.1	1.00	1.3	0.63
Standardization	**3.9**	**0.01**	1.7	0.35	4.7	0.05	0.7	0.63
Flexibility	**3.9**	**0.02**	0.9	1.00	1.0	1.00	**3.1**	**0.04**
Productivity	**4.2**	**0.01**	1.2	1.00	1.2	1.00	**3.4**	**0.02**
Reactivity to changes	**5.5**	< **0.01**	0.9	1.00	1.0	1.00	2.5	0.12
Platform independence	**9.9**	< **0.01**	0.8	0.75	3.5	0.22	**4.3**	**0.01**

Table 5: Effects of additional factors on benefit achievement rate.

3.3. RQ2: benefits achievement

As a preliminary step we evaluate the cross correlations among different benefits. Table A.7 (in appendix) reports the correlation (in terms of Pearson's ϕ) among the different benefit achievements; in bold the statistically significant correlations. We observed one strong correlation (between *Flexibility* and *Reactivity to changes*), 11 moderate correlations, 18 small, and 6 negligible. The strong correlation is a clue indicating a possible common shared underlying construct, as discussed later in section 5.

In order to address RQ 2, first of all we look at which benefits are concretely achieved by using models, then we look into the factors affecting the verification of those benefits: basic modelling vs. MD*, individual key MD* techniques, and additional co-factors.

3.3.1. Overall

Table 4 (in the first six columns) reports the frequency of achieved advantages among the 105 modelling users. The table also reports the estimated benefit achievement ratio and the relative 95% CI, as percentages. The benefits are sorted from the most likely (*Design*) to the less one (*Platform independence*). According to the likelihood categories defined in section 2.5.2 we can classify two benefits as *very likely*: usefulness for design assessment (*Design support*) and documentation improvement (*Documentation*). In addition, we found that improved comprehension during maintenance (*Maintenance*), higher product quality (*Quality*), and improved standard compliance (*Standardization*) are *likely* benefits. In the *Possible* category fall *Flexibility*, *Productivity*, and *Reactivity to changes*. Eventually, *Platform independence* can be considered as *Unlikely*.

3.3.2. MD adopters vs. Basic Modellers*

On the left-hand side of Figure 5, we report the benefit achievement ratio separately for users of MD* techniques and those of basic modelling. From the diagram we can appreciate that for most benefits the difference between more advance users (MD*) and basic users (basic modelling) is limited. The odds ratios of the achievement ratios are reported in the sixth column of table 4 aside the p value of the Fisher test. The odds ratios are almost all greater than 1, indicating an improvement of the achievement ratio for MD* adopters, the exception being documentation benefits that appear to be less likely achieved among MD* adopters.

From the test results we can infer that a significant difference exists only for three benefits: *Standardization*, *Productivity*, and *Platform independence* (they are marked with a circle in Figure 5). MD* adopters are two and half times more likely to achieve standardization benefits compared to basic modelling users, five and half times more likely to achieve productivity benefits, and five times more likely to achieve platform independence. We observe that the above differences are consistent with the different intended purposes of modelling and MD*: communication the former and code generation or execution the latter. For these three benefits, we classified the likelihood of benefits for MD* adopters; the only benefit that could be classified in a different category was *Platform independence*: considering all modellers it was considered *Unlikely*, while restricting to MD* adopters it becomes *Possible*.

3.3.3. Key MD techniques*

More in detail, on the right-hand side of Figure 5, we can observe the achieved benefits divided by adopters of code generation, model interpretation, and model transformation, respectively.

Table 4 reports (six rightmost columns) the odds ratios of achieving benefit for adopters of specific MD* key techniques vs. non adopters. Code generation induces one significant difference, concerning *Productivity* which is 3.9 times more likely to be achieved as a benefit when the technique is adopted. Most significant

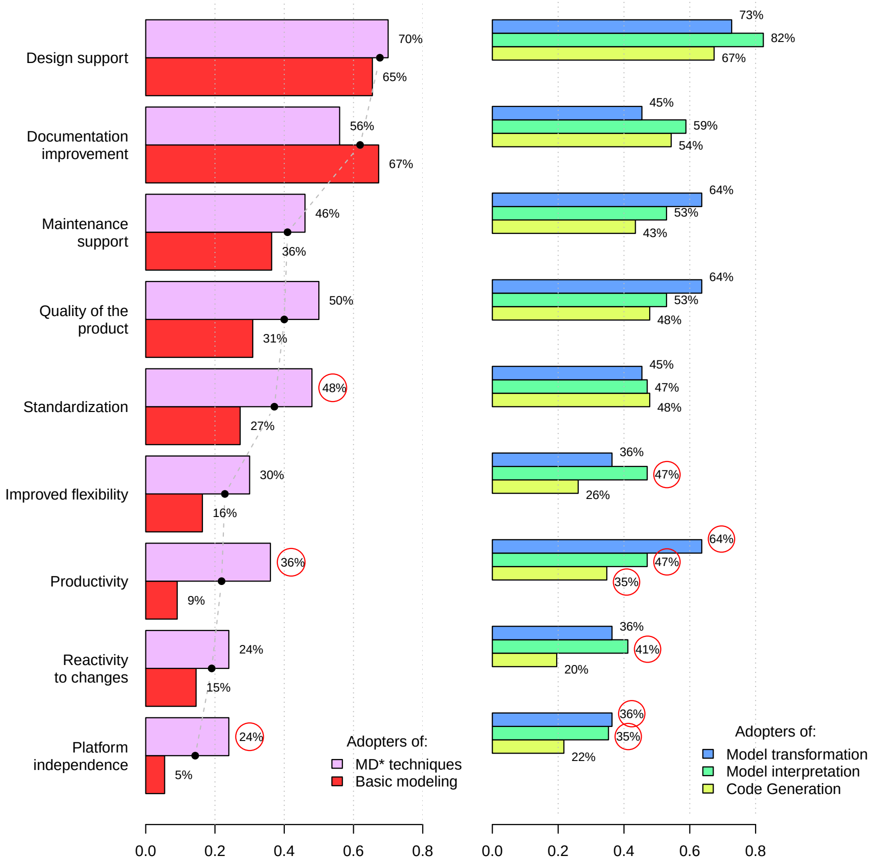


Figure 5: Benefits achieved. By "Basic modelling" we mean use of models not resorting on any MD* technique. Circles indicated statistically significant difference.

differences in benefit achievement are observed when model interpretation is applied: *Flexibility*, *Productivity*, and *Reactivity to changes* benefits are circa four times more likely to be achieved when interpretation is used, in addition *Platform independence* if almost five times more likely. When model transformation is applied, we observe a eight times increment in *Productivity* achievement likelihood, and a four times increment for *Platform independence*.

The most notable spike is observable in the right-hand diagram of Figure 5 and concerns *Productivity* benefit achieved by the adopters of model transformation techniques; it corresponds to the odds ratio of eight we described above.

3.3.4. Company size and additional factors

We verified whether a correlation exists between the company size categories and the achievement ratio of each individual benefit. The χ^2 test did not reveal any significant relationship, all p values being greater than 10%.

As a last step of analysis of the achieved benefits we focus on additional factors: toolsmithing, use of UML and UML profiles, and the adoption of DSLs. Table 5 reports the benefit achievement odds ratios for the presence of the additional factors vs. their absence.

We can observe that respondents who developed to some extent their tools (toolsmithing) had a significantly higher likelihood of achieving several benefits: four times higher for *Standardization*, *Flexibility*, and *Productivity* benefits; more than five times higher for *Reactivity to changes* and almost ten times higher for *Platform independence*. Apparently the adoption of UML and UML Profiles is not linked to any increased benefit achievement ratio. The adoption of domain specific languages (DSLs) is linked to three times higher likelihood of achieving *Flexibility* and *Productivity* benefits, and four times higher chances to achieve *Platform*

		Occurrence ratio			Basic		MD*		MD* vs. Basic	
	Freq	Ratio	95% CI	Relevance	Num	Prop.	Num	Prop.	OR	p
Too much effort required	67	50%	41% .. 58%	Relevant	23	46%	21	60%	1.75	0.27
Not useful enough	64	47%	39% .. 56%	Relevant	26	52%	15	42%	0.70	0.51
Lack of competencies	46	34%	26% .. 43%	Relevant	16	32%	10	28%	0.85	0.81
Refusal from management	33	24%	18% .. 33%	Moderately Relevant	14	28%	9	25%	0.89	1.00
Lack of supporting tools	23	17%	11% .. 25%	Moderately Relevant	9	18%	3	8%	0.43	0.34
Refusal from developers	19	14%	9% .. 21%	Scarcely Relevant	9	18%	6	17%	0.94	1.00
Inadequacy of supporting tools	14	10%	6% .. 17%	Scarcely Relevant	4	8%	8	22%	3.36	0.06
Cost of supporting tools	14	10%	6% .. 17%	Scarcely Relevant	6	12%	4	11%	0.95	1.00
Fear of lock in	13	10%	5% .. 16%	Scarcely Relevant	2	4%	7	20%	5.87	**0.03**
Not flexible enough	10	7%	4% .. 14%	Scarcely Relevant	4	8%	3	8%	1.08	1.00

Table 6: Problems encountered preventing adoption of MD*.

independence.

3.4. RQ3: problems

As a preliminary step we assess the cross correlation among the different problems. Table A.8 reports the correlation coefficients (Pearson's ϕ) for each pair of problems; in bold are reported the statistically significant correlations. We observe no strong correlation, three moderate ones, and 14 small ones. Most of the correlations (28) are negligible.

3.4.1. Overall

Table 6 reports the frequency of each problem and the corresponding occurrence ratio. According to the 95% confidence interval (columns 4 and 5) we can assign a relevance category conforming to the thresholds defined in section 2.5. We observe three relevant potential problems: *Too much effort required*, *Not enough expected usefulness*, and *Lack of competencies*.

3.4.2. Sometimes vs. Never

Figure 6 reports the frequency of problems preventing the adoption of models (as reported by respondents never using modelling) or hindering it (respondents using modelling just sometimes) is reported. We remind that participants that use models always were not asked this question.

We looked at the odds ratios of problem occurrence for respondent that never adopt modelling vs. those who sometimes adopt, all the values are close to 1 – between 0.65 and 1.71 – except for *Refusal from developers* and *Inadequacy of supporting tools* having with odds ratio 0.41 and 0.26 respectively. Indicating they are the most likely reasons for not adopting modelling.

We could not observe any statistically significant difference between the problem occurrence ratio of modellers vs. non modellers.

3.4.3. Basic Modellers vs. MD* adopters

Table 6 also reports the frequency and occurrence ratio of each problem among basic modelling adopters vs. MD* technique users. In addition (rightmost two columns) the odds ratio of incurring in a problem for MD* vs. basic modellers is reported together with the relative statistical significance, computed by means of the Fisher test. The only statistically significant difference concerns the *Fear of lock-in*, which appears almost six times more frequent among the MD* techniques adopters. Next to this difference, though not significant, is the one about the *Inadequacy of supporting tools*.

3.4.4. Company size

Figure 7 reports the distribution of problems occurrence ratio by company size. We can observe a substantially uniform distribution among the different size classes, with the exception of a few cases. According to the χ^2 test, we identified two problems whose occurrence is significantly related to the company size: *Lack of competencies* and *Refusal from developers*. Concerning the former, small and micro companies are surprisingly reporting this problem less than medium, large, and individual companies. *Refusal from developers* is instead mostly reported in medium sized companies.

4. Discussion

The three research questions pinpoint three aspects that we wish to recap here: diffusion/relevance, advantages, and problems of modelling and MD*.

The first result (RQ1.1) we obtained from the survey is that of a large **diffusion** of modelling practices (68% of respondents) and a relatively ample diffusion of MD* techniques (48% of the adopters of modeling and 32% of the entire valid sample). Considering our plain relevance criteria, we can classify modeling as a highly relevant technique in the industrial context; while MD*

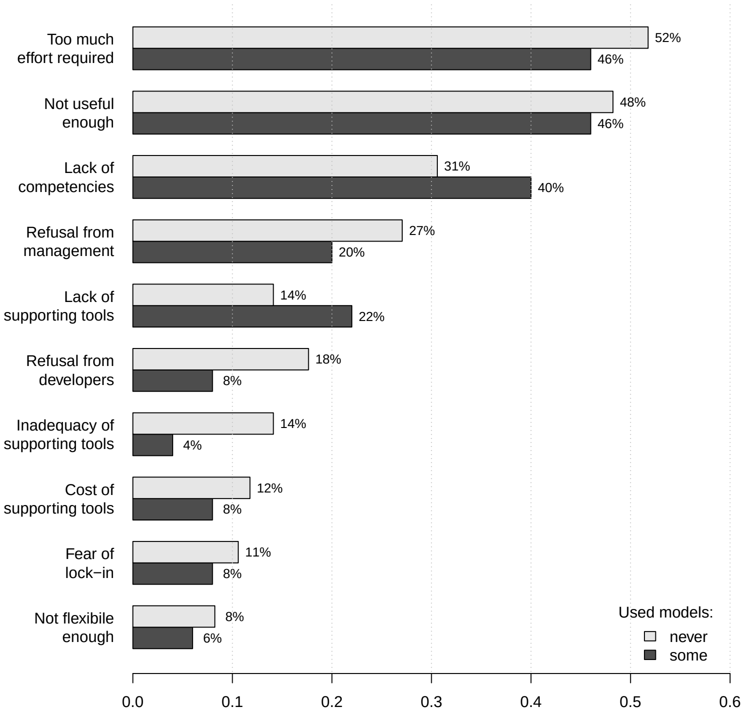


Figure 6: Prevalence of problems limiting adoption of modeling.

can be considered as relevant. This information is important per-se for us researchers: it means that research conducted in this context has the potential to yield a significant impact on practitioners.

When looking at how the adoption is distributed with respect to company size (RQ1.2), we observe a bimodal shape both for modelling (see Figure 3) and MD* (Figure 4): medium-large companies are more keen to adopt modelling and MD*, small companies are less prone to these practices, and micro and individual companies are similar to large ones in this respect.

As far as key MD* techniques are concerned, almost all adopters of MD* do apply code generation, one third apply model interpretation, and one fifth use model transformations. The adoption rate of code generation (Figure 4) by micro companies[11] is significantly higher than other companies while the two latter techniques are largely adopted by micro companies only. The large diffusion of those techniques in micro companies is in stark contrast to the very low adoption in small companies.

Our explanation for the above facts is that model interpretation and transformation are relatively novel techniques, at least more advanced than code generation which is well-known and more used in the industry. As a consequence, their adoption brings risks: apparently only micro companies, and to a much lesser extent medium-large ones, are willing to take them. Micro companies are possibly driven by the competition to stay in the market, medium-large ones perhaps believe in the advanced techniques as competitive advantages. Moreover, micro companies can afford to adopt, more easily, not fully mature solutions — nowadays, provided in the market for Model transformation and interpretation — which are not easily accepted in larger companies. In addition, the “micro” size allows them more flexibility in using new technologies and processes (one possible co-cause for that could be the larger freedom developers have in micro companies). Finally, probably, in large companies there is more resistance, by developers, to the introduction of novel techniques and processes than in smaller ones.

Another important finding concerns the likelihood of **benefits** achievement (RQ2). The adoption of modelling makes improved design and documentation benefits very likely to be achieved (see Table 4), while maintenance, quality, and standardization are simply likely.

Figure 8 shows the the relationship between MD*

---

[11] $2 \leq \text{size} \leq 10$

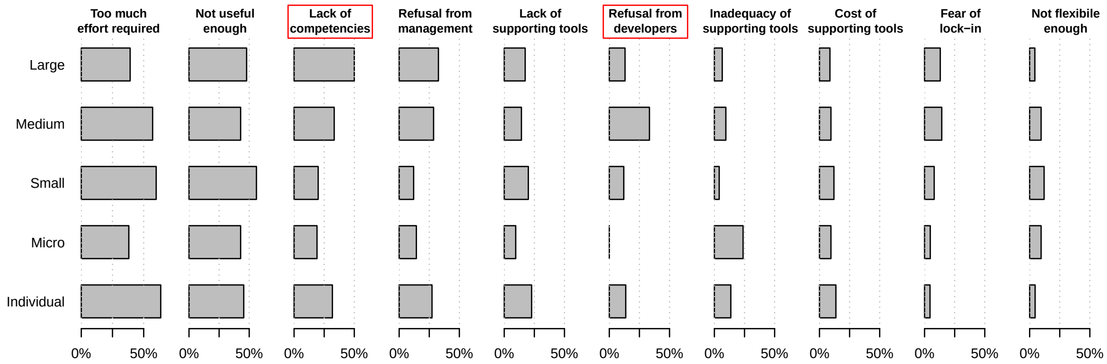


Figure 7: Problem occurrence ratio per company size category.

techniques and the benefits: each checkmark indicates that the adoption of the technique in the column causes a statistically significant improvement in the likelihood of achieving the benefit in that row.

In practice the four topmost rows in Figure 8 represent the most commonly achievable benefits. *Support in design definition*, *Improved documentation*, easier *Maintenance*, and higher *Quality* are obtained through the simple adoption of modelling (RQ2.1). We could not find statistically significant evidence of any MD* effect on their achievement because, due to the design of our survey, we could not compare modellers vs. non modellers. (Table 4).

MD* specific techniques play a significant role for the remaining five benefits investigated in our study (RQ2.2). The odds of achieving *Flexibility* is almost four times higher when model interpretation is adopted. When it comes to *Productivity* all the three MD* techniques increase the likelihood obtaining an improvement, in particular model transformation may increase the odds by eight times. *Reactivity to changes* is easier to achieve when model interpretation is adopted, while the chances of achieving *Platform independence* are increased by applying model transformation or model interpretation.

Moreover, Figure 8 shows how other two techniques – toolsmithing and DSLs –, usually associated with MD* practices, can play a significant role. The development of own tools (toolsmithing) is a significant enabler for all bottom five benefits in the figure (*Flexibility*, *Productivity*, *Reactivity to changes*, and *Platform independence*). Particularly relevant is the contribution of toolsmithing to *Platform independence*: an Odds ratio of 10 is very high and suggests that projects having platform independence among their priorities should seriously consider building their own tools. Unfortunately, we have no information about the effort required to realize those tools. More evidence is needed to drive the "make or buy" decision. Finally, the use of Domain Specific Languages increases the odds of achieving *Flexibility*, *Productivity*, and *Platform independence*, by three to four times.

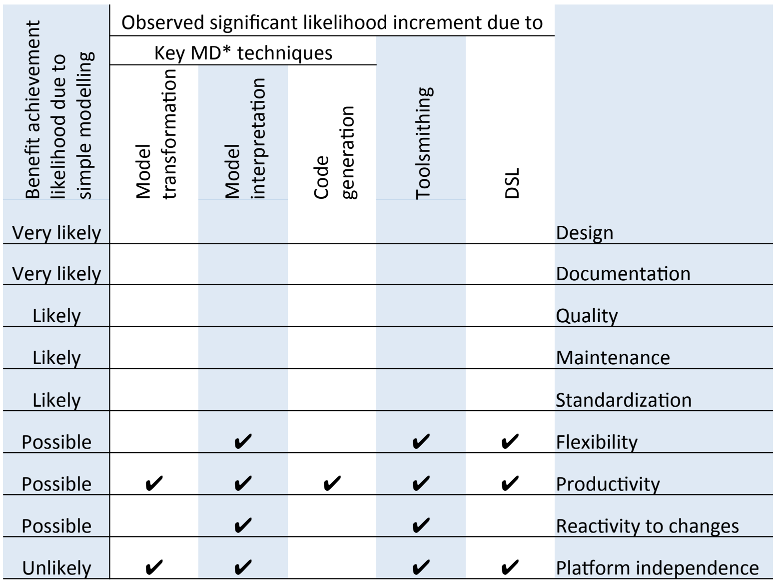
We can interpret statistical significance as a causal relationship, which represents the empirical basis for pragmatic decision making. Under such perspective, Figure 8 illustrates the factors that can play the role of the deal breaker in achieving a given benefit, on the basis of the collected empirical evidence.

In practice, Figure 8 is an attempt to synthesise a piece of evidence that can be leveraged by practitioners. For instance, if the goal for a project is *Productivity*, all the three MD* specific techniques – code generation, model interpretation and model transformation – can help; while if we aim at achieving *Reactivity to changes* only model interpretation and toolsmithing can help.

We are not claiming the solutions derivable by Figure 8 are the only possible ones: they are combinations that proved statistically significant in our sample. As such, they represent the starting point in finding a customized solution for a goal or set of goals.

Another insight we get from a overall glance at Figure 8 is that MD* and other techniques play an important role where the simple modelling is weaker and vice-versa. From this picture we confirm the impression that simple modelling and MD* are two complementary sets of techniques.

If we look at the **problems** restraining from the adoption of modelling (RQ3), as reported by respondents, we observe that most problems are cited less often by participants that never use modelling than those who use it sometimes (see Figure 6). The possible explanation for



Benefit achievement likelihood due to simple modelling	Observed significant likelihood increment due to					
	Key MD* techniques					
	Model transformation	Model interpretation	Code generation	Toolsmithing	DSL	
Very likely						Design
Very likely						Documentation
Likely						Quality
Likely						Maintenance
Likely						Standardization
Possible		✔		✔	✔	Flexibility
Possible	✔	✔	✔	✔	✔	Productivity
Possible		✔		✔		Reactivity to changes
Unlikely	✔	✔		✔	✔	Platform independence

Figure 8: Achievable benefits with Modelling and MD* techniques adoption effects.

this difference is that potential problems and risks are considered more often when development teams has to repeatedly balance pros and cons for using modelling or MD*.

The notable exception to the above trend is represent by the *Lack of competencies* and *Lack of supporting tools*. Such problems appear to be the main showstoppers preventing altogether the adoption of modelling and MD*.

The *Fear of lock-in* is a problem which seem to affect a lot more MD* practitioners than adopters of simple modelling. This could derive from the lack of affirmed standards in MD* while in modelling UML seems to be widely used. This lack results into both poor options for replacing tools by equivalent alternatives and problems in building heterogenous tool-chains.

We believe that the findings of our survey, and in particular, these two above problems (*Lack of competencies* and *Lack of supporting tools*) deserve attention from Italian industries and universities. The former should invest more in research, tools building (software modelling and MD* tools are needed) and training (experts in MD* are needed), and the latter should produce more experts in modeling and model driven techniques. This strongly suggests to improve university curricula with specific courses dealing with topics related to software modelling, and more specifically with code generation, model execution, and model transformation. Most of the times, students are trained to build new systems using traditional processes and only in the better case the foundation of MD* are explained in software engineering courses (e.g., this is the case in the Università di Genova — Italy). While it is our opinion that they should focus more on modelling and model driven techniques (in particular in automatic code generation, given that, it is the most used in the industry). On the university side, the *Lack of supporting tools* and dissatisfaction about them (Figure 6) should be a prompt to produce new prototypes and experiment more in this direction. On the industrial side, we can infer a huge market opportunity for modeling and MD* tools. Moreover, investments in this market could obtain large returns especially for large companies. And possibly, companies and universities should collaborate to make better tools.

5. Threats to validity

We analyse the potential threats to the validity of our study according to the four categories suggested in [45]. In general on-line surveys are considered to have lower internal validity and stronger external validity in respect to other means of empirical investigations as case-studies or experiments [36].

Construct validity threats concern the relationship between theory and observation. They are mainly related to the measurements performed in the study. In particular there are two distinct issues: (i) whether we measured the usage of techniques in the right way (measurement instrument and process) and, (ii), whether we selected the right attributes to represent the construct of technology usage (measured attribute).

Concerning the measurement instrument – a personal opinion survey – the items definitions and the scales used to code the answers are key factors, and they can potentially influence our results because of the difficulty respondents could have encountered in either understanding or (mis)interpreting the items. We paid particular attention to mitigate such threats:

- The questionnaire was designed using a standard approach, trying to avoid as much as possible ambiguous questions [21].
- We strived to formulate the items in a simple and straightforward style (see Table 1)
- We inserted the meaning of relevant terms (e.g., model, model execution, MD*) in the questionnaire (directly in the questions or as footnotes).
- We guaranteed assistance by phone and e-mail to respondents, to support them in case of unclear questions.
- We conducted a pre-test of the instrument, by means of three pilot studies with industrial professionals, to check that the questions were understandable, before putting on-line the questionnaire.
- In particular the lists of advantages (question Dev11) and problems (question Dev09) proposed to the respondents could have been incomplete. In the first case, we opted for a closed question (see Figure 2) on the basis of the expert practitioners judgement in the pilot. On the contrary, in the second case, since the range of problems is potentially very large, we opted for a semi-open question: we provided a set of predefined options but let the respondent free to add others (see Figure 2). Considering the answers of Dev9, we can confirm that the list of proposed problems was quite complete, since respondents sparingly used the “free” option.

As far as the measured attributes are concerned, we identified the MD* usage construct with the adoption of three key techniques: code generation, model transformation, and model interpretation. We are confident that they represent the main features of MD*, or at least a largely shared view of MD*, although we cannot exclude there exists some community with a different perspective on MD*.

In addition from the correlation analysis, we found a strong correlation between *Reactivity to changes* and *Flexibility*; this may indicate the possibility of a single construct underlying the two measures. We explored the possibility of removing one of the two, but eventually we preferred to retain all the information and keep both at the expense of a small additional complexity of the study. We believe that this decision does not invalidate the conclusions of our study.

Internal validity threats concern confounding factors that may affect the outcome of our results. In general, it is hard to control these factors. It is well-known that, a survey being an unsupervised study, the level of control is very low.

Internal validity is mainly threatened by coverage issues:

- We incurred in a possible selection bias due to the self-exclusion of participants not interested in modelling. Self-exclusion is a well-known problem especially in Internet surveys advertised by means of mailing lists and groups. The possible threat consists in an over estimation of the proportion of respondents who declared interest in modelling and therefore of the overall relevance of modelling and MD* in the Italian industry. The main impact of this issue would be on the answer to RQ1, the findings relative to the other research questions should not be significantly affected.

  We tried to mitigate this threat by presenting the request as a study on software development without emphasizing the modelling aspect and addressing it to a varied population. Though we cannot definitely rule out the threat, there are two aspects in the collected data that make us at least confident that the magnitude of the threat is limited. First, a significant number of valid respondents never used modeling therefore a portion of the sampled population although not interested in modelling did filled in the survey anyway. Second, among the incomplete answers – respondents who started the questionnaire but did not complete it and were thus discarded from analysis – (26 occurrences), 10 respondents did anyway provide a response to question DEV08: all of them affirmed to use models *sometimes* (8 respondents) or *always* (2 respondents); if in a conservative move we ascribe the remaining 16 to the group of respondents *never* using modelling, the picture we obtain considering all the 181 (155 complete + 26 incomplete) responses is not significantly different than the complete responses alone.

- Another threat derives from the possible "foreign units" in the sample: the target population of our study consisted of development teams, it is possible that the questions were answered by a responded without the required knowledge (e.g., the secretary of the IT manager). We addressed this concern in the protocol: we explicitly required the questionnaire to be filled in by technical personnel involved in the development. Even in the case of a knowledgeable respondent, (s)he could be unaware of some details; this is more likely if the team is very large [21].
- Finally, the sampling procedure made possible to select duplicate units: two different members of the same development team could have answered our questionnaire. We addressed this threat by means of a post-survey validation: we found that the respondents from the same company actually worked in distinct business units and belonged to distinct teams.

Conclusion validity threats concern the possibility to derive illegitimate conclusions from the observations. When hypothesis testing was used to compare two or more populations, we adopted non-parametric tests (Kruskal-Wallis, Fisher exact test and Mann-Whitney), that can be used without specific assumptions (e.g., without checking data normality). Similarly, we used the proportion test to determine the confidence interval for relevance of modelling in general.

External validity threats concern the extent to which our findings can be generalized. For our survey, we used the Commerce Chamber database to render our sample as representative as possible. Had we used only this sampling source, we could have performed a stratified sampling, using strata based on company size as was performed e.g. in [8]. However, we decided to integrat that sample using a non-probabilistic sampling schema (the same was done in [33]). As a result, the solution we selected is cost-effective and allowed us to obtain a sample large enough to achieve a reasonable number of adopters of all the techniques. This should be considered interpreting the results we obtained: even if the demographics of our sample is quite diverse, the generalization of our results to the entire population may not be appropriate. Moreover, given the sampling strategy we adopted, we cannot calculate the response rate (this problem is common in software engineering surveys [21]). We are also aware that the size of our sample is not large enough for generalization purposes (of course, further data points will be highly desired to better generalize our findings). However, that size is similar to other industrial surveys conducted on different software engineering subjects (e.g., [14, 19, 20, 25, 41]).

6. Related work

Literature reports some anecdotal evidence collected through case-studies (e.g., [26]), while rigorous empirical studies evaluating software modelling and MD* are quite rare. Carver et al. [4] performed a literature review considering the most common venues where articles related to MD* are published. They noted that the 73% of the papers they considered didn't contain any form of validation, consequently they affirm that the rigor of empirically validated research in software modeling is rather weak and the community need to focus more on this aspect. In particular, as stated in [29], there are a few reports on the advantages of applying MDE in industry, thus more empirical studies are needed to strengthen the evidence. Also van Deursen et al. [7] report on the importance of having more empirical studies, in particular about the improvements produced by MD* adoption on maintainability costs.

In the rest of this section, we focus on empirical studies about issues, challenges and benefit of modelling and MD* adoption. Coherently with the stance adopted in our survey, we decided to avoid a clear partitioning between software modelling and MD*, also given that the two aspects are often interwoven.

We grouped the related studies as much as possible considering their category: literature reviews, surveys, case studies, and experience reports.

6.1. Literature reviews

Budgen et al. [3] conducted a systematic literature review on UML. Authors underline the necessity of more empirical studies about the adoption of UML. Most of the empirical studies are laboratory experiments while more field studies are needed.

Mohagheghi and Dehlen in their work [29] introduce a literature review of empirical studies (and more in general of industrial experiences), from 2000 to 2007, about MDE in industry. The goal is evaluating MDE benefits and possible limitations. They selected 25 papers and their main conclusions are: (i) MDE is applied in a wide range of domains, (ii) MDE can lead to various benefits (i.e., higher productivity and improved communication), (iii) MDE is not considered mature enough and there are no appropriate tool chains, and (iv) quantitative evidence was found in one paper only, about productivity gains. Our findings are consistent with theirs, especially with items (ii) and (iii).

6.2. Surveys

Forward et al. in their work [9] analyses the results of a survey with 113 software practitioners (about two-thirds were from Canada or the USA) on the perception of software modeling. They mainly investigate how, when and why software developers use (or not) models and which notations and tools are adopted. In their sample, modelling is performed at least sometimes by over 95% of participants (in our case the modellers are 68%). Similarly to us, they try to answer the following research question:"Why do some developers prefer not to model?" (similar to our RQ3). They report that the biggest problem is the synchronization between models and code (models become out of date with code). We do not have evidence of this problem. Other problems are the quality of the generated code and issues with the modelling tools (e.g., too expensive, "heavyweight" and difficult to use). Our results are in line with this latter perception. A portion of our sample think that modelling tools represent a limitation for MD* and more in general for software modelling. The main findings reported in their conclusion are: (i) developers consider models in a broader sense (i.e., not only UML models but also textual DSL models), (ii) UML is the predominant modelling notation (similar to a result we reported in a previous article [42]) but is often used informally, (iii) modelling tools are mainly used for documentation (a fact that could explain the large percentage of documentation benefits reported in Figure 5), and (iv) it is uncommon that models are used for generating code.

Davies et al. in [6] report the result of a survey conducted in Australia on the status of conceptual modelling that has received 312 responses. This study aims to determine the actual modelling practice, giving a answer to the research question:"how do practitioners actually use conceptual modelling in practice?" that is specified by three sub-questions: (i)"which are the tools and techniques used for conceptual modelling?" (ii)"what is the purpose of modelling?", and (iii) "what are the major problems and benefits specific to conceptual modelling?" The last sub-question is similar to ours RQ2 and RQ3, but from a different perspective. They have identified problems and benefits in the usage of conceptual modelling by means of textual analysis of data relative to problems and perceived key success factors. Concerning the last sub-questions they report which are the factors influencing the continued use. The major key factor is relative advantage/usefulness, other factors (in order of relative importance) are: communication to/from stakeholders, internal knowledge of techniques, user expectations management, understanding the model integration into the business, tool/software deficiencies. The first factor corresponds to our finding of a commonly achieved benefit (documentation improvement), the last factor corresponds to one of the problem we found limiting adoption of modelling (inadequacy of supporting tools).

Hutchinson et al. in [18] report the results of an empirical study on the assessment of MDE in industry. Their work has two goals: identify the reasons of success or failure of MDE and understand how MDE is actually applied in industry. They employed three forms of investigation: questionnaires, interviews, and on site observations, having as target practitioners, MDE professionals and companies practising MDE respectively. The questionnaire has received over 250 responses from professionals (the most of them are working in Europe). Some of the reported findings are: (i) about two-thirds of the respondents believe that using MDE is advantageous in terms of productivity, maintability and portability, (ii) the majority of respondents use UML as modelling language, and a good number use in-house developed DSLs, (iii) almost three quarters of respondents think that an extra training is necessary to use MDE, (iv) the majority of respondents agree that code generation is an important aspect of MDE productivity gain, and (v) a little less than half of the respondents think that MDE tools are too expensive. We observed similar perceptions in our survey except for the issue of extra-training which was not considered in our survey, however we observed that the lack of competencies is one of the problems most frequently reported by companies. Differently from the results of their survey, the cost of supporting tools is seen as a problem only by a small proportion of respondents in our sample.

Nugroho and Chaudron in their work [33] analyses the results of a survey about the UML usage and its perceived impact on quality and productivity with 80 professional software engineers. The findings reveals that: (1) the majority of the respondents agreed that a model should describe parts of a system that are more critical and complex, instead of specifying all parts of a system equally; (2) incompleteness in UML models is associated to implementation problems (it brings to deviations in the implementation) and, (3) using UML impacts productivity in analysis, design and implementation but not in maintenance. This last finding seems partially in contrast with our results where easier maintenance is one of the benefits associated with modelling. Similarly to our survey, their findings reveal problems associated to the usage of tools: the features in current UML CASE tools that should help maintaining aligned code and design, e.g., reverse engineering and round-trip engineering, are not yet mature.

6.3. *Case studies*

The work of Hutchinson, Rouncefield and Whittle [17] focuses on the deployment of MDE in industry. It illustrates three industrial case studies in three different business contexts (printer, car manufacturing and telecommunication companies) and identifies some lessons learned. In particular, the importance of complex organizational, managerial and social factors in the success or failure of the MDE deployment. The authors report some organizational factors that can affect the success or the failure of MDE deployment. The factors that can affect it positively are: (i) a progressive and iterative approach, (ii) user motivation in the MDE approach, (iii) an organizational willingness in integrating MDE in the whole organization, and (iv) having a clear business focus (where MDE is adopted as a solution for new projects). Instead, factors that can affect it negatively are: (i) the decision of adopting MDE being taken by IT managers, in top-down fashion and implemented "all at once", (ii) MDE being imposed on the developers, and (iii) an inflexible organization with a lack of integration of MDE in previous processes. The only common aspect with the work proposed in [17] concerns the motivation of developers. The corresponding finding lies in the problems reported in Figure 6 (refusal from developers and refusal from management).

Mohagheghi et al. [31] interviewed – using convenience sampling – developers from four companies involved in an initiative called MODELPLEX. They examined the factors affecting adoption of MDE. Regarding usefulness they found uncertain results: most participants recognize the usefulness of models but they are not sure about the impact on the quality of the final product or the effects on productivity. MDE is perceived as not simple: its complexity makes it viable for engineers but not for non technical people. This finding is confirmed by our results reported in [42, 40]. They show that only in a few cases business experts are involved during modelling tasks. Regarding compatibility with the existing development process the companies complained about the lack of standards and the consequent lock-in effect. All interviewed companies reported some problems in integrating their existing approaches with MDE. Tools could have been part of their problems, them being not considered satisfying by a part of the sample. In particular, some participants expressed several concerns about the scalability of the MDE approach to large projects. Advantages reported are limited to the usefulness for documentation and communication purposes. Major reasons preventing adoption of MDE are the immaturity of tools and processes as well as the lack of competencies. Such latter conclusions are largely consistent with our findings.

6.4. *Experience reports*

Heijstek et al. [15] study the impact of MDD on a large scale industrial project and the main features of a large scale industrial MDD project. They produced an experience report using, as sources of information, data from the Subversion repository and semi-structured interviews with team members. About the impact of MDD, the conclusions are that: almost two-thirds of the total effort is spent on developing models and that the team members report an increase in productivity, besides a perception of improvement of the overall quality and a reduction of complexity. The authors confirmed the increase of quality by counting the average number of defects w.r.t. the average number of defects found in similarly sized projects in which MDD was not used. Their findings – including the improvement of the final product – have been observed also in our survey. While we do not have data about the effort spent on realizing the models, many participants considered that effort to be too big and therefore affecting their decision to adopt modelling. They identified two typical features of large scale MDD projects: (i) high average complexity per diagram, (ii) activity diagrams and class diagrams are the more used UML diagrams.

Mohagheghi et al. [30] report a list of challenges and success criteria of MDE adoption. They, summarize them after having conducted two real projects in large organizations. The most important challenge is the definition of a MDE environment, that require the company: (i) to develop a communication language for technical and domain experts by means of UML profiles and/or meta-models and (ii) to select and integrate tools for building, transforming, storing, reusing and composing models. This last point is particularly difficult to reach due to the lack of such a tool-chain on the market. Such finding is consistent with our results. They also report that training is a major challenge (same problem stated in [18]). In addition, they see advantages in the gradual introduction of MDE in the industry and in the creation of expert teams to support and create tools.

7. Conclusions and future work

In this paper we presented some results from a survey performed to investigate: (1) what is the relevance of software modelling and MD* in the Italian industry, (2) what are the attainable benefits from such techniques, and (3) which are the problems impeding the adoption of modelling and MD*.

First, we found that the practice of producing models is quite widespread (68% of the entire sample), while the proportion of development teams using MD* techniques is smaller (48% of the adopters of modeling) but still noteworthy. Among the MD* techniques, the most used is code generation (almost all adopters of MD* do apply code generation). Considering our relevance criteria, we can classify modelling as a *highly relevant* technique in the Italian industrial context while MD* can be considered as *relevant*. This result implies that any research gain in this field has the potential of a large return on practice.

Second, more relevant benefits such as: *Support in design definition*, *Improved documentation*, easier *Maintenance*, and higher *Quality* seem to be obtained when simple models are used and no further improvement is observed with MD* adoption. On the other hand, MD* plays a significant role for *Productivity*, *Platform independence* and *Standardization*. The complete set of empirically backed causal relationships (technique adopted → benefit) observed in our study is shown in Figure 8. Such figure is an attempt to synthesise pieces of evidence that can be used by practitioners to decide whether to invest on a specific technique, given the desired benefits.

Third, the main problems mentioned by adopters of modelling mimic the typical anecdotal ones: models require too much effort to be produced and often they are not useful enough. Instead, the problems preventing altogether the adoption of both modelling and MD* seems to be related to a mix of technological and human factors, that is lack of supporting tools and lack of competencies. Such findings can originate suggestions useful for both Italian companies and universities. The former should invest more in research and tools building in order to address the lack of supporting tools (and in general tools inadequacy) reported by the respondents of our survey. The latter should produce more experts in modelling and model driven techniques so to raise the level of expertise and satisfy today's industrial needs.

As future work, we would like to compare the level of adoption of modelling and MD* in Italian companies to the situation in other countries by replicating this study in other nations. In particular, we are interested to understand whether the companies in other nations have the same problems that the Italian have and whether (or not) they achieve the same benefits using modelling and MD*.

Acknowledgements

We would like to thank the people who dedicated their time to take part in our survey and the organizers of CodeMotion for the help provided to diffuse our survey.

References

[1] R. Acerbis, A. Bongio, M. Brambilla, S. Butti, Webratio 5: An eclipse-based case tool for engineering web applications, in: L. Baresi, P. Fraternali, G.J. Houben (Eds.), Web Engineering, volume 4607 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, 2007, pp. 501–505.

[2] A. Agresti, An Introduction to Categorical Data Analysis, Wiley-Interscience, 2007.

[3] D. Budgen, A.J. Burn, O.P. Brereton, B.A. Kitchenham, R. Pretorius, Empirical evidence about the uml: a systematic literature review, Softw. Pract. Exper. 41 (2011) 363–392.

[4] J.C. Carver, E. Syriani, J. Gray, Assessing the frequency of empirical evaluation in software modeling research, in: M. Chaudron, M. Genero, S. Abrahão, P. Mohagheghi, L. Pareto (Eds.), Proceedings of the First Workshop on Experiences and Empirical Studies in Software Modelling, volume 785 of *CEUR Workshop Proceedings*, 2011, pp. 20–29.

[5] S. Ceri, P. Fraternali, A. Bongio, Web modeling language (WebML): a modeling language for designing web sites, Computer Networks 33 (2000) 137–157. Elsevier North-Holland, Inc., New York, NY, USA.

[6] I. Davies, P. Green, M. Rosemann, M. Indulska, S. Gallo, How do practitioners use conceptual modeling in practice?, Data & Knowledge Engineering 58 (2006) 358–380. Elsevier Science Publishers B. V., Amsterdam, The Netherlands.

[7] A. van Deursen, E. Visser, J. Warmer, Model-Driven Software Evolution: A Research Agenda, Technical Report, TUDelft-SERG, 2007.

[8] E. Egorova, M. Torchiano, M. Morisio, Actual vs. perceived effect of software engineering practices in the italian industry, Journal of Systems and Software 83 (2010) 1907 – 1916. Elsevier Science Inc., New York, NY, USA.

[9] A. Forward, O. Badreddin, T.C. Lethbridge, Perceptions of software modeling: A survey of software practitioners, in: 5th Workshop from Code Centric to Model Centric: Evaluating the Effectiveness of MDD, 2010, pp. 12–24.

[10] M. Fowler, R. Parsons, Domain-Specific Languages, Addison-Wesley, 2011.

[11] R. France, B. Rumpe, Model engineering, Journal Software and Systems Modeling 2(2) (2003) 73–75. Springer-Verlag Heidelberg, Heidelberg, Germany.

[12] R.M. Groves, F.J.J. Fowler, M.P. Couper, J.M. Lepkowski, E. Singer, R. Tourangeau, Survey Methodology, John Wiley and Sons, 2009.

[13] V. Gurov, M. Mazin, A. Narvsky, A. Shalyto, Tools for support of automata-based programming, Programming and Computer Software 33 (2007) 343–355. MAIK Nauka/Interperiodica distributed exclusively by Springer Science+Business Media LLC.

[14] O. Hauge, Open Source Software in Software Intensive Industry - A Survey, Technical Report, Norwegian University of Science and Technology Department of Computer and Information Science, 2007. Http://daim.idi.ntnu.no/masteroppgaver/IME/IDI/2007/3290/masteroppgave.pdf.

[15] W. Heijstek, M.R.V. Chaudron, Empirical investigations of model size, complexity and effort in a large scale, distributed model driven development process, in: Proceedings of the 2009 35th Euromicro Conference on Software Engineering and Advanced Applications, SEAA '09, IEEE Computer Society, Washington, DC, USA, 2009, pp. 113–120.

[16] B. Hjørland, F. Sejer Christensen, Work tasks and socio-cognitive relevance: a specific example, Journal of the American Society for Information Science and Technology 53 (2002) 960–965.

[17] J. Hutchinson, M. Rouncefield, J. Whittle, Model-driven engineering practices in industry, in: Proceedings of the 33rd International Conference on Software Engineering, ICSE '11, ACM, New York, NY, USA, 2011, pp. 633–642.

[18] J. Hutchinson, J. Whittle, M. Rouncefield, S. Kristoffersen, Empirical assessment of MDE in industry, in: Proceedings of the 33rd International Conference on Software engineering, ICSE '11, ACM, New York, NY, USA, 2011, pp. 471–480.

[19] J. Hutchinson, J. Whittle, M. Rouncefield, S. Kristoffersen, Empirical assessment of mde in industry, in: Proceedings of the 33rd International Conference on Software Engineering, ICSE '11, ACM, New York, NY, USA, 2011, pp. 471–480.

[20] A. Jelitshka, M. Ciolkowski, C. Denger, B. Freimut, A. Schlichting, Relevant information sources for successful technology transfer: a survey using inspections as an example, in: Proceedings of the International Symposium on Empirical Software Engineering and Measurement (ESEM), IEEE, 2007, pp. 31–40.

[21] B. Kitchenham, S. Pfleeger, Personal opinion surveys, in: F. Shull, Singer (Eds.), Guide to Advanced Empirical Software Engineering, Springer London, 2008, pp. 63–92.

[22] A.G. Kleppe, J. Warmer, et al., MDA Explained: The Model Driven Architecture: Practice and Promise, Addison-Wesley Longman Publishing Co., Inc, 2003.

[23] M. Leotta, F. Ricca, M. Ribaudo, G. Reggio, E. Astesiano, T. Vernazza, An Exploratory Survey on SOA Knowledge, Adoption and Trend in the Italian Industry, in: Proceedings of 14th International Symposium on Web Systems Evolution (WSE 2012), IEEE, 2012, pp. 21–30.

[24] T.C. Lethbridge, A survey of the relevance of computer science and software engineering education, in: Proceedings of the 11th Conference on Software Engineering Education and Training, IEEE Computer Society, Washington, DC, USA, 1998, pp. 56–66.

[25] J. Li, R. Conradi, O. Petter, N. Slyngstad, M. Torchiano, M. Morisio, C. Bunse, A state-of-the-practice survey on risk management in development with off-the-shelf software components, IEEE Software 34 (2008) 271–286.

[26] A. MacDonald, D. Russell, B. Atchison, Model-driven development within a legacy system: an industry experience report, in: Australian Software Engineering Conference, IEEE Computer Society, 2005, pp. 14–22.

[27] S. Mellor, M. Balcer, Executable UML: A foundation for model-driven architecture, Addison-Wesley, 2002.

[28] S. Mellor, A. Clark, T. Futagami, Model-driven development - guest editor's introduction, Software, IEEE 20 (2003) 14–18.

[29] P. Mohagheghi, V. Dehlen, Where is the proof? - a review of experiences from applying MDE in industry, in: Proceedings of the 4th European conference on Model Driven Architecture: Foundations and Applications, ECMDA-FA '08, Springer-Verlag, Berlin, Heidelberg, 2008, pp. 432–443.

[30] P. Mohagheghi, M. Fernandez, J. Martell, M. Fritzsche, W. Gilani, MDE adoption in industry : Challenges and success criteria, Models in Software Engineering (2009) 54–59.

[31] P. Mohagheghi, W. Gilani, A. Stefanescu, M. Fernandez, An empirical study of the state of the practice and acceptance of model-driven engineering in four industrial cases, Empirical Software Engineering (2012) 1–28.

[32] H. Motulsky, Intuitive biostatistics: a nonmathematical guide to statistical thinking, Oxford University Press, 2010.

[33] A. Nugroho, M.R. Chaudron, A survey into the rigor of uml use and its perceived impact on quality and productivity, in: Proceedings of the ACM-IEEE international symposium on Empirical software engineering and measurement, ESEM '08, ACM, New York, NY, USA, 2008, pp. 90–99.

[34] Object Management Group, Business Process Model and Notation, v. 2.0, Standard, Object Management Group, 2011.

[35] Object Management Group, Unified Modeling Language, Superstructure, v. 2.4, Specifications, Object Management Group, 2011.

[36] T. Punter, M. Ciolkowski, B. Freimut, I. John, Conducting on-line surveys in software engineering, in: Proceedings of the International Symposium on Empirical Software Engineering, (ISESE), 2003, pp. 80–88.

[37] D.C. Schmidt, Guest editor's introduction: Model-driven engineering, Computer 39 (2006) 25–31.

[38] B. Selic, The pragmatics of model-driven development, IEEE Software 20 (2003) 19–25.

[39] J. Singer, M. Storey, D. Damian, Selecting empirical methods for software engineering research, in: F. Shull, Singer (Eds.), Guide to Advanced Empirical Software Engineering, Springer, 2008, pp. 285–311.

[40] F. Tomassetti, M. Torchiano, A. Tiso, F. Ricca, G. Reggio, Maturity of software modelling and model driven engineering: a survey in the italian industry, IET Seminar Digests 2012 (2012) 91–100.

[41] M. Torchiano, M. Di Penta, F. Ricca, A. De Lucia, F. Lanubile, Migration of information systems in the italian industry: A state of the practice survey, Information and Software Technology 53 (2011) 71–86.

[42] M. Torchiano, F. Tomassetti, F. Ricca, A. Tiso, G. Reggio, Preliminary findings from a survey on the MD* state of the practice, in: International Symposium on Empirical Software Engineering and Measurement (ESEM), IEEE, 2011, pp. 372 –375.

[43] M. Voelter, Best practices for DSLs and model-driven development, Journal of Object Technology 8 (2009) online at http://www.jot.fm.

[44] D.S. Walonick, Survival Statistics, StatPac, Inc., 1997.

[45] C. Wohlin, P. Runeson, M. Höst, M. Ohlsson, B. Regnell, A. Wesslén, Experimentation in Software Engineering - An Introduction, Kluwer Academic Publishers, 2000.

Appendix A. Cross correlations

We report here the cross correlations among the benefits in Table A.7 and this among the problems in Table A.8.

	Design	Documentation	Maintenance	Quality	Standardization	Flexibility	Productivity	Reactivity to changes
Documentation	**0.38**							
Maintenance	**0.33**	**0.21**						
Quality	**0.23**	-0.08	**0.35**					
Standardization	0.07	0.08	0.16	**0.22**				
Flexibility	0.18	-0.13	**0.24**	**0.39**	0.14			
Productivity	0.17	-0.06	0.12	**0.37**	**0.26**	**0.42**		
Reactivity to changes	0.02	-0.12	**0.24**	**0.35**	**0.28**	**0.60**	**0.39**	
Platform independence	0.17	0.10	**0.21**	0.17	**0.25**	**0.36**	**0.38**	**0.36**

Table A.7: Benefits achievement correlation.

	Not useful enough	Too much effort required	Lack of supporting tools	Inadequacy of supporting tools	Lack of competencies	Fear of lock in	Refusal from developers	Refusal from management	Cost of supporting tools
Too much effort required	0.01								
Lack of supporting tools	0.04	-0.02							
Inadequacy of supporting tools	0.07	0.10	0.04						
Lack of competencies	0.10	-0.12	**0.21**	0.06					
Fear of lock in	0.04	0.08	-0.15	-0.03	0.03				
Refusal from developers	0.09	0.15	-0.01	0.07	0.11	0.08			
Refusal from management	**0.18**	0.02	0.15	0.15	**0.32**	-0.01	**0.22**		
Cost of supporting tools	0.02	0.15	0.04	**0.36**	-0.04	**0.38**	0.07	0.03	
Not flexibile enough	0.01	0.12	0.02	0.09	-0.08	**0.20**	0.13	0.04	**0.18**

Table A.8: Correlation of potential problems.