A Systematic Review of the Use of Requirements Engineering Techniques in Model-Driven Development

Grzegorz Loniewski, Emilio Insfran, Silvia Abrahão

ISSI Research Group, Department of Computer Science and Computation
Universidad Politécnica de Valencia
Camino de Vera, s/n, 46022, Valencia, Spain
grlo@ei.upv.es, einsfran@dsic.upv.es, sabrahao@dsic.upv.es

Abstract. Model-Driven Development (MDD) emphasizes the use of models at a higher abstraction level in the software development process and argues in favor of automation via model execution, transformation, and code generation. However, a current challenge is how to manage requirements during this process whilst simultaneously stressing the automation benefits. This paper presents a systematic review of the current use of requirements engineering activities in MDD processes and their actual automation level. 65 papers from the last decade have been reviewed from an initial set of 877 papers. The results show that although MDD techniques are used to a great extent at platform-independent models, platform-specific models, and code level, at the requirements level most MDD approaches use only partially defined requirements models or even natural language. We additionally identify several research gaps such as a need for more efforts to explicitly deal with requirements traceability and providing a better tool support.

Key words: model-driven development, requirements engineering, systematic review

1 Introduction

Software engineering experiences show that in recent decades model-based development of systems has become an essential factor in reducing costs and development time. Furthermore, properly managed, well-documented and easily understandable software requirements definitions have a great impact on final product quality [4]. However, requirements engineering (RE) is one of the software engineering disciplines in which model based approaches are still not widespread. Requirements are generally regarded as text fragments that are structured to a greater or lesser extent and which are interpreted by stakeholders and developers, who manually manage the requirement interrelationships [22].

A variety of methods and model-driven techniques have been published in literature. However, only a few of them explicitly include the requirements discipline in the Model-Driven Development (MDD) process. This paper presents a review of scientific papers published in the last decade which include the use of RE techniques in the context of an MDD process. In order to provide a balanced and objective summary of research evidence, a systematic literature review (SLR) process is considered as an appropriate method to carry out such a review in software engineering [2].

Various systematic literature reviews have been performed in the field of RE, such as that of Davis *et al.* [4] which presented an SLR on the effectiveness of requirements elicitation techniques. However, RE techniques focused on model-driven methodologies and processes are rarely researched. An attempt to describe a combination of these two fields of software engineering can be seen in the work of Nicolás and Toval in [29] in which they show the influence of models on the generation of textual requirements. To date no literature reviews for requirements models applied to the model-driven environment have been found.

Model-driven development is often applied to Platform-Independent Models (PIMs) and Platform-Specific Models (PSMs), but is hardly ever found in Computation-Independent Models (CIMs). In order to investigate this issue in greater depth, we present a systematic literature review with the main goal of studying the use of requirements engineering techniques in MDD processes and their actual level of automation. The systematic review presented in this paper was started in December 2009 and was later updated in February 2010.

This paper is organized as follows. Section 2 describes the protocol followed to carry out the systematic review. Section 3 presents the results obtained. Section 4 discusses the threats to the validity of the results. Finally, Section 5 presents our conclusions and suggests areas for further investigation.

2 Research method

We follow the approach proposed by Kitchenham [14] for systematic literature review. A systematic review is a means of evaluating and interpreting all the available research that is relevant to a particular research question, topic area, or phenomenon of interest. It aims at presenting a fair evaluation of a research topic by using a trustworthy, rigorous, and auditable methodology. A systematic review involves several stages and activities, which are briefly explained below:

- **Planning the review**: the need for the review is identified, the research questions are specified and the review protocol is defined.
- **Conducting the review**: the primary studies are selected, the quality assessment used to include studies is defined, the data extraction and monitoring is performed and the obtained data is synthesized. In this stage we added a new activity to test the reliability of the review protocol.
- **Reporting the review**: dissemination mechanisms are specified and a review report is presented.

The activities concerning the planning and conducting phases of our systematic review are described in the following subsections. The report on the review stage is presented in Section 3.

2.1 Research question

In order to examine the current use of requirements engineering techniques in model-driven development and their actual level of automation we formulate the following research question: "What requirements engineering techniques have been employed in model-driven development approaches and what is their actual level of automation?". This research question will allow us to derive the gathering process of the current knowledge about requirements engineering techniques in MDD and to identify gaps in research in order to suggest areas for further investigation. The review has been structured following the PICOC criteria [31]:

- **Population:** Research papers presenting MDD processes and techniques,
- **Intervention**: Requirements engineering methods and techniques,
- **Comparison**: Analysis of all approaches based on the specified criteria,
- **Outcome**: Not focused on achieving any specific result,
- **Context**: Research papers based on RE techniques used in MDD.

Our review is more limited than a full systematic review as suggested in [14] since we did not follow up the references in papers. In addition, we did not include other references such as technical reports, working papers and PhD thesis documents. This strategy has been used in another systematic reviews conducted in software engineering field [25].

2.2 Sources selection

Two types of searching methods were used to select appropriate and representative papers in the field of requirements and model-driven engineering. The first type, automatic searching, was based on four main sources of scientific papers databases: IEEE *Xplore* (IEEE), ACM Digital Library (ACM), Science Direct (SD), and SpringerLink (SL). In addition, a manual search was done in the following representative conferences and journals: Requirements Engineering conference (RE), the Conference on Model Driven Engineering Languages and Systems (MODELS), and Requirements Engineering Journal (REJ).

This manual search method was applied to check the correctness of the automatic review and to have a closer look at the works published in these sources exploring new trends and approaches.

2.3 Identifying and selecting primary studies

Based on the research goal, a search string was created to identify primary studies. The search string consists of three parts: the first part linking the works that describe requirements engineering techniques using models, the second part that refers to model-driven engineering concepts, and finally the third part to describe the relation between requirements and other models of the MDD process. We experimented with several search strings and the following retrieved the greatest amount of relevant papers:
(requirements engineering OR requirements-based OR requirements-driven OR

requirements model OR business model OR CIM OR Computation Independent Model) ***AND*** *(MDA OR MDE OR model-driven OR model-based OR model*)* ***AND*** *(transform* OR traceability)*. The concrete syntax of this search string was adapted to each digital library we used.

2.4 Inclusion criteria and procedures

The searching configuration included limitations to the type of papers and content. Papers that had been taken into consideration were only those that are research papers presenting approaches to MDD-based requirements engineering or software development process with requirements traceability. Moreover, only papers published in conferences/workshops proceedings and scientific journals between Jan, 1999 and Jan, 2010 were considered as significant for the research. The following types of papers were excluded:

- papers describing model-driven principles without describing a concrete requirements engineering technique,
- papers presenting requirements engineering techniques that are not related to model-driven principles,
- books, tutorials, standards definitions, poster publications,
- short papers (papers with less than 4 pages),
- papers not written in English.

2.5 Data extraction strategy

The data extracted were compared according to the research question stated, which is here decomposed into the following criteria that are described below.

- **Type of requirements** (criterion 1) can be of two types: *software* requirements which are requirements describing only functionalities of software under development and *business* requirements which include the information not only about functionality of the future system, yet also the business context, organizational structure of the enterprise, processes, etc. which not necessarily will be a part of the system to be developed.
- Next, the information about the type of **requirements structure** (criterion 2) is collected. Requirements can be represented as *models* (two types of models are distinguished: *standard* models expressed in the only modelling language that is considered as a standard (UML from OMG) and *non-standard* other type of a model). Requirements can also be expressed in *natural language* or other type of textual or graphical representation.
- In case of using models as requirements specification, the information about **type of models** (criterion 3) is gathered. These models can be: *structural*, *behavioural*, *functional* or of *other* type.
- Model **transformations provided** (criterion 4) is an interesting topic which concentrates on an important amount of research work.

- **Level of transformations** (criterion 5), as proposed in Mens *et al.* [26], for transformations we can also analyze the languages in which are expressed, both source and target models and their abstraction level. In this case transformations are classified as *endogenous* and *exogenous*, depending on the relation of abstraction level between these models.
- **Use of transformation languages** (criterion 6) is also analyzed. Transformations can be defined by *standard* languages such as QVT or ATL or by *non-standard* transformation languages. [1]
- **Transformations automation level** (criterion 7). We consider a transformation as *automatic* if the entire process of obtaining the target model can be done without participation of the transformation user. Next, we distinguish *interactive* or semi-automatic approaches and finally the *manual* approach.
- **Requirements traceability** (criterion 8). Requirements traceability refers to the ability to describe and follow the life of a requirement, in both a forwards and backwards direction. We focus on the *post-requirements specification* (post-RS). Post-RS refers to those aspects of a requirement's life that result from inclusion in a requirements specification as defined by Gotel and Finkelstein [9]. The reviewed papers were analyzed to study the traceability to and from *analysis*, *design*, and *implementation* artifacts.
- **Traceability automation** (criterion 9) is investigated giving us some conclusions on the effort needed to manage the traceability within the MDD process.
- **Tool support** (criterion 10). We analyzed if there is a tool that performs the MDD transformations on requirements and generates model(s), also if it provides the possibility of requirements traceability and monitoring of this traceability during the whole software development process.
- **Type of validation** (criterion 11) conducted. Three different types of strategies can be carried out depending on the purpose of the validation and the conditions for empirical investigation [7]: *survey*, *case study*, and *experiment*. A survey is an investigation performed in retrospect, when the method has been in use for a certain period of time. A case study is an observational study where data is collected for a specific purpose throughout the study. An experiment is a formal, rigorous, controlled investigation. The set of validation methods does not include a theoretical examples of proof-of-concepts.
- Finally, the **actual usage** (criterion 12) of the requirements engineering technique is analyzed. The paper is classified as *industrial* if it presents a requirements engineering technique which was proposed (and is being used) in the context of a company. Otherwise, it is classified as *academic* if the paper describes an academic environment and no evidence about its current usage is provided.

[1] We include ATL in the standard category since it is widely used in academia and can be considered as a *de facto* standard.

2.6 Conducting the review

The conducting stage possesses the following activities: selection of primary studies, data extraction, and data synthesis. How and from which sources the primary studies were selected was described in the previous sections. Based on this information, the research resulted in 867 potentially related papers significant for the research topic.
In addition, the manual search which includes the RE and MODELS conferences and the REJ journal resulted in 10 relevant papers related to the research topic.

Table 1 presents the results of the final set of relevant papers selection for each of the sources. *Searching results* row shows the number of papers obtained from each source that resulted from the search string and the manual search. *Finally selected* row indicates the number of papers included to review after the rejection of papers that satisfied at least one of the exclusion criteria or if the paper topic was not adequate to the systematic revision purpose. Furthermore, duplicated papers were discarded, taking into consideration the first digital library where they appear, the newest publication, or the most complete one.

As a result 65 papers were chosen for the final review. First, the research was done in December 2009 and then updated by Jan, 31, 2010. A complete list of reviewed papers is available at:
`www.dsic.upv.es/~einsfran/review-remdd.htm`.

Table 1. Number of the review results

	Automatic				Manual	
Source	IEEE	ACM	SD	SL	RE/MODELS/REJ	Total
Searching results	163	641	24	39	10	877
Finally selected	21	25	9	7	3	65

3 Results

This section discusses the results of the review commenting each criterion. Table 2 shows a summary of number of papers obtained as a result of the review. This table is organized into groups regarding the selection criteria and the publication sources.

The results for the **requirements type** (criterion 1) show that the majority of works (62%) focus on *software requirements* (e.g., Insfran *et al.* in [11] where requirements are represented through the use of the Techniques for Requirements and Architecture Design (TRADE) such as mission statement, function refinement tree, and use cases). In this context the Service Oriented Architecture (SOA) has gained a significant popularity in recent years. Various works

Table 2. Systematic review results

Selection criteria		Sources					Total
		IE	ACM	SD	SL	MS	
1 Requirements type	Software	14	13	5	6	2	40
	Business	7	12	3	2	1	25
2 Requirements structure	Standard model	5	9	4	1	2	21
	Non-standard model	9	9	2	1	1	22
	Template	0	1	0	0	0	1
	Structured natural language	6	7	2	0	0	15
	Natural language	2	2	1	3	0	8
	Other	0	0	1	1	0	2
3 Type of models	Structural	3	4	1	1	0	9
	Behavioural	8	15	5	2	3	33
	Functional	2	2	0	0	1	5
	Other	2	1	2	0	0	5
4 Transformations provided	Yes	19	19	8	4	2	52
	No	2	6	1	3	1	13
5 Transformations level	Endogenous	4	4	0	1	0	9
	Exogenous	16	17	8	3	2	46
6 Standard transformations	Yes	1	4	0	0	0	5
	No	16	14	6	4	2	42
7 Transformations automation	Automatic	7	11	3	0	2	23
	Interactive	4	3	2	1	0	10
	Manual	9	8	2	2	0	21
8 Traceability provided	To analysis	8	4	2	5	0	19
	To design	3	7	1	1	0	12
	To implementation	4	6	1	0	1	12
	None	12	14	5	2	2	35
9 Traceability automation	Automatic	5	5	3	2	1	16
	Manual	4	6	0	2	0	12
10 Tool support	Traceability only	1	3	1	3	1	9
	Transformation only	5	6	4	2	2	19
	Traceability and transformation	1	0	1	0	0	2
	None	14	15	3	2	0	34
11 Type of validation	Survey	0	0	0	1	0	1
	Case study	11	10	2	2	3	28
	Experiment	0	1	0	1	0	2
	None	10	14	7	3	0	34
12 Approach scope	Academic	14	19	7	5	1	46
	Industry	7	6	2	3	2	20

describe automation methods for services specification, e.g., Jamshidi [12] proposes a novel method called ASSM (Automated Service Specification Method) to automatically specify the architecturally significant elements of service model work product from the requirements. Only 38% of reviewed papers use *business requirements* as a base for further development. At this point it is worth mentioning that many approaches use the i* notation to describe these business requirements. For example, Mazón *et al.* [24] introduce the use of the i* framework to define goal models for data warehouses and automatic conceptual modelling. There also exist other approaches for generating UML models from business requirements like the one of Raj *et al.* [32]. Their approach presents an automated transformation of business designs from SBVR Business Design (Semantics of Business Vocabulary and Rules) to UML models which bridges the gap between business people and IT people.

The results for the **requirements structure** (criterion 2) show that from the reviewed papers 62% of those that apply some RE techniques in the MDD approach used models as a means to represent the requirements. Herewith two types of models are distinguished: UML *standard models* (30%) (the most frequently used are class, activity, and sequence diagrams) and *non-standard models* (32%) such as goal, aspect, feature, or task-based requirements models. Other alternatives to represent the requirements are: i) *structured natural language* (22%) where requirements are described in a way easy to analyze. For example, Mauco *et al.* [23] use a natural language oriented model which models the vocabulary of a domain by means of the Language Extended Lexicon (LEL); ii) *natural language* (12%), for example, Fliedl *et al.* in [8] where they propose the use of sophisticated tagging in the requirements lexical analysis; iii) *templates* (1%) (e.g. the requirements traceability approach by Cleland-Huang *et al.* in [3]) and finally, iv) *other* type of specifications mostly proprietary domain specific (3%). These results are shown in Figure 1(A).

The results for the **type of models** (criterion 3) show that from those approaches that use models for requirements specification the most frequently used type of model is *behavioural* model (63%). In many works, this type of model is used as use case specifications (e.g. [19], [10] and [34]) or very often also as goal models (e.g. in [18], [17] or [16]). Other alternatives less used are: *structural* (17%), *functional* (10%) (e.g. activity or sequence UML diagrams), and *other* types of models (10%) such as Requirements-Design Coupling (RDC) model proposed by Ozkaya *et al.* [30] or Stakeholder Quality Requirement Models described with semantic web technology in [1]. The summary of the aforementioned results are shown in Figure 1(B).

The results for the **transformations from requirements phase** (criterion 4) show that a total of 80% of the papers reviewed describe different types of transformations from requirements specifications. We can distinguish different types of transformations such as mappings, transformations using patterns and templates, transformations implemented in transformation languages (QVT, ATL), linguistic operations on textual requirements or graph transformations, etc. On the other hand 20% of papers do not provide such transformations (see Figure 1(C.a)) focusing the approach on other aspects of MDD such as traceability of requirements, e.g. Cleland-Huang *et al.* [3].

The results for the **level of model transformations** (criterion 5) give an outcome about the abstraction level of source and target models in the transformation process (see Figure 1(B.b)) according to the previously introduced classification presented by Mens *et al.* in [26]. The great majority of approaches (83%) transform a source model into a different type of model (*exogenous* transformations). Changing the target model specification language or abstraction level in relation to the source models (mainly goal models or natural language scenarios descriptions) is mostly used in works that apply UML models as a target model. For example, Debnath *et al.* in [5] describe natural language requirements transformation to UML class model. Also many approaches that start the MDD process from business requirements specifications propose exoge-

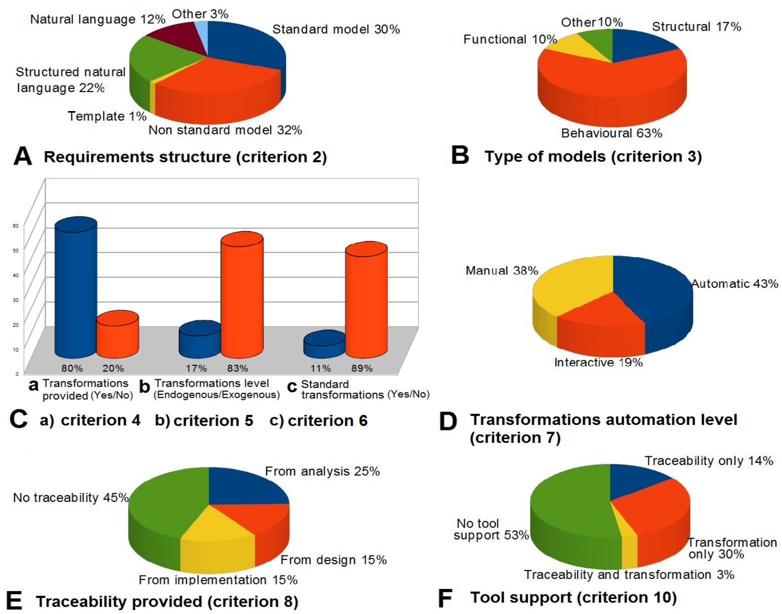

Fig. 1. Results for some representative criteria of the systematic review

nous transformations, for example, in [32] where the business requirements that are specified with the Semantics of Business Vocabulary and Rules (SBVR) are transformed into UML models (other examples could be [13] and [38]). Some works provide transformations of models within the same modelling language, yet on different abstraction level, e.g. transforming UML use case diagrams into UML activity diagrams. *Endogenous* transformations apply in 17% of reviewed approaches. For instance, this type of transformations are used by Laguna and Gonzalez-Baixauli in [16] where endogenous transformations are considered as requirements configurations used to validate the completeness and consistency of the initial requirements configurations represented as a set of goal and feature models. Another approach with this kind of transformations was used in validation of scenarios presented by Shin *et al.* in [36].

The results for the **use of transformation languages** (criterion 6) show that 89% of transformations included in this systematic review use different kinds of specific mappings, refinements or patterns based transformations or languages other than standardized transformation languages (see Figure 1(C.c)), e.g. Raj *et al.* [32] define some specific mapping rules to transform business design to UML models. Only 11% of the works use one of the two considered in this work as standard languages: QVT and ATL. For example, in [15] Koch *et al.* propose transformations based on QVT as transformation language for Web system design.

The results for the **transformations automation level** (criterion 7) show the current state of automation for the transformations defined in the MDD process. 43% of the approaches perform fully *automatic* transitions from requirements specifications to analysis and design models. For example, Zhang and Jiang in [38] propose a well-defined mapping of requirements defined in Visual Process Modeling language (VPML) at the CIM level to Business Process Execution Language (BPEL) description at the PIM level. Following, 19% of the papers describe *interactive* or semi-automatic transformation methods, e.g. [17] or [21]; 38% of the papers discuss *manual* transformations, e.g. [16]. Figure 1(D) shows a summary of the results for this criterion.

The results for the **traceability** (criterion 8) show the support for requirements traceability in the reviewed papers. Regarding the classification of traceability from [9], this work focuses on post-RS traceability, which is the relationship between requirements specification (RS) artifacts and analysis as well as design artifacts. This traceability can be forward and backward traceability. Since the work deals with the model-driven environment, the majority of approaches that possess model transformations assume that there exists forward traceability, although not all of these approaches have explicit mechanisms for that traceability support. 45% of works do not provide backwards traceability, although forward traceability is possible. Such situation can be found in the approach described by Insfran in [11] where the forward traceability is implicitly assumed by the transformation rules provided, yet there is no reference to the backward traceability. Moreover, 25% of works provide backward traceability from the analysis phase (e.g. in [17] where goal models are transformed to statecharts and the backward traceability is recorded); 15% of works provide traceability from design and implementation phases mainly from user interface prototypes and test case implementations (e.g., Sousa *et al.* in [37] present an approach for requirements tracing from user interface implementation). In addition, some authors like Naslavsky *et al.* in [28] describe complete solutions for tracing products from different phases to requirements (specified as scenarios). These results are shown in Figure 1(E).

The results for the **traceability automation** (criterion 9) show that more than half of the methods (57%) that have some traceability support provide an automated tool for the traceability management, e.g. [28]. This means that in these approaches the effort needed to manage the requirements traces within the MDD process is quite low or none. The number of manual traceability approaches is also significant: 43%. For example, in the work of Sousa *et al.* [37] user interfaces can be logically linked to the business processes defined at the requirements level.

The results for the **tool support** (criterion 10) for the MDD approach show that from the reviewed papers, as expected, not even half of them have a tool support. With respect to those approaches that have some process automation tools the following categories are distinguished: 30% of approaches have tool support for model transformations, e.g. [38] where a tool for automatic BPEL models creation is supported basing on the source model; 14% of works only support traceability, e.g. in [3] Cleland-Huang *et al.* propose a traceability of requirements specification without any previous model transformations; and finally, only 3% of the papers describe technological solutions including both transformations and traceability support. A work of Rash *et al.* [33] could serve as a good example here since provides R2D2C transormations as well as includes the traceability support. Retrieving models and formal specifications from existing code is also possible.

On the other hand, 53% of works do not offer any tool support, however, most of them emphasize such a necessity and point it out as a part of their future work. These results are shown in Figure 1(F).

The results for the **type of validation** (criterion 11) give an overview of the evaluation methods used in selected papers. Three validation methods were taken into account in order to classify the results: survey, case study, and experiment. 52% of reviewed papers do not present any validation performed and a more or less detailed example was the only mean provided to ilustrate the feasibility of the approach. Case studies, more or less well-defined, were used in 43% of the cases. The majority of papers, especially those describing academic researches, use theoretical example (e.g. [20]), whereas industrial researches but not only, were very often evaluated with a *case study* (e.g. [35]). It is also worth noticing that controlled *experiments* were used only in 3% of the works (e.g. [6]), and a validation by surveys were used in only 2% of the papers.

Finally, the results for the **actual usage** (criterion 12) show that 70% of selected papers were defined in an *academic* context and 30% were defined in an *industrial* setting. The predominance of the academic proposals found in this review shows that new approaches to deal with techniques for modelling, transformations, and processes, that include RE as a part of the MDD process in industrial contexts are still needed. One representative attempt from industry to apply model-driven RE in its own development process is the AQUA project [6].

4 Threats to validity

This section discusses the threats to validity that could affect the results of the systematic review. We have validated our review protocol to ensure that the research was the most correct, complete, and objective as possible. Regarding the source selection we have selected four digital libraries (IEEEXplore, ACM, Science Direct, and Springerlink) containing a very large set of publications in the Software Engineering field. The search string was defined trying different combinations of terms extracted from Requirements Engineering and model-driven techniques papers. We also applied patterns for searching terms and adopted the search string to advanced methods of source searching in each digital library selected. In this way the reproducibility of the automatic searching results would be possible.

Possible limitations of this study concern publication bias, publication selection, inaccuracy in data extraction, and misclassification. Publication bias refer to the problem in such a way that positive results are more likely to be published than negative results [14]. We try to alleviate this threat, at least to some extent, by scanning relevant conference proceedings and journals. In addition, the digital libraries contain many relevant journal articles and conference proceedings. With regard to publication selection we chose the sources where requirements engineering and model-driven engineering works were normally published. However, we did not consider grey literature (e.g. industrial reports, PhD thesis) or unpublished results.

With regard to the search string we tried to collect all the strings that are representative of the research question. We refined the search string several times in order to obtain the maximum of papers related to the systematic review. In addition, we have taken under consideration synonyms and and have included the lexeme of words.

We attempted to alleviate the threats of inaccuracy in data extraction and misclassification by conducting the classifications of the papers with two reviewers. The discrepancies among the evaluations were discussed and a consensus was found.

5 Conclusions and future work

In this paper we have presented a systematic review on the use of requirements engineering techniques in MDD processes.

Research in the last decade has shown that surprising progress can be observed with regard to how much precision and automatic support can be applied to models in the early RE process, although a complete solution supporting MDD methodologies is still lacking [27].

The results of our research are in line with the aforementioned findings. They show that there is no complete solution containing a tool support with which to manage the models in the requirements phase and to make further use of them in an automatic manner. Many MDD methodologies include some requirements engineering activities but these are hardly ever included in the mainstream of the automated process. Moreover, this systematic revision verifies that models are not exclusively used to describe requirements in the MDD context to serve as the input for model-driven transformations, but that templates, graphs, structured and non-structured natural language are also used. To date automated model transformations appear to be poorly used since the majority of them are designed as different kinds of mappings and graph transformations that use non-standard transformation languages. Greater benefits could be obtained by applying model-driven transformations at the requirements level [27].

After analyzing the results of our systematic review we can draw the conclusion that models are relatively rarely used in the MDD context, and that the post requirements specification traceability is not well defined. In addition there are no complete solutions that are well documented and evaluated, which also influences so many poor attempts to include RE in the MDD process, especially in the industrial setting. There are no empirical studies on the benefits of the use of RE techniques in the improvement of productivity, efficiency and software development process quality and effectiveness.

Although our findings may be indicative of the field, further investigation is needed to confirm the results obtained. As a future work we plan to extend this study and maintain it updated by including other bibliographical sources of scientific and industrial papers. Moreover, this systematic review is a part of a more extensive research work which has the main goal of proposeing a new methodology for an automated RE method for MDD processes. This work is being developed in the context of a project that is founded by the Ministry of Science and Innovation in Spain.

References

1. Biffl, S., Mordinyi, R., Schatten, A.: A model-driven architecture approach using explicit stakeholder quality requirement models for building dependable information systems. In: WoSQ '07: Proceedings of the 5th International Workshop on Software Quality. p. 6. IEEE Computer Society, Washington, DC, USA (2007)
2. Brereton, P., Kitchenham, B.A., Budgen, D., Turner, M., Khalil, M.: Lessons from applying the systematic literature review process within the software engineering domain. J. Syst. Softw. 80(4), 571–583 (2007)
3. Cleland-Huang, J., Hayes, J.H., Domel, J.M.: Model-based traceability. In: TEFSE '09: Proceedings of the 2009 ICSE Workshop on Traceability in Emerging Forms of Software Engineering. pp. 6–10. IEEE Computer Society, Washington, DC, USA (2009)
4. Davis, A., Dieste, O., Hickey, A., Juristo, N., Moreno, A.M.: Effectiveness of requirements elicitation techniques: Empirical results derived from a systematic review. In: RE '06: Proceedings of the 14th IEEE International Requirements Engineering Conference. pp. 176–185. IEEE Computer Society, Washington, DC, USA (2006)
5. Debnath, N., Leonardi, M., Mauco, M., Montejano, G., Riesco, D.: Improving model driven architecture with requirements models. In: ITNG 2008: Fifth International Conference on Information Technology: New Generations, 2008. pp. 21 –26 (april 2008)
6. Escalona, M.J., Gutiérrez, J.J., Rodríguez-Catalán, L., Guevara, A.: Model-driven in reverse: the practical experience of the aqua project. In: EATIS '09: Proceedings of the 2009 Euro American Conference on Telematics and Information Systems. pp. 1–6. ACM, New York, NY, USA (2009)
7. Fenton, N.E., Pfleeger, S.L.: Software Metrics: a rigorous and practical approach. International Thompson computer, 2nd edn. (1996)
8. Fliedl, G., Kop, C., Mayr, H.C., Salbrechter, A., Vhringer, J., Weber, G., Winkler, C.: Deriving static and dynamic concepts from software requirements using sophisticated tagging. Data & Knowledge Engineering 61(3), 433 – 448 (2007), advances on Natural Language Processing - NLDB 05
9. Gotel, O.C.Z., Finkelstein, C.W.: An analysis of the requirements traceability problem. Proceedings of the First International Conference on Requirements Engineering pp. 94–101 (April 1994)
10. Hinchey, M.G., Rash, J.L., Rouff, C.A., Gracanin, D.: Achieving dependability in sensor networks through automated requirements-based programming. Computer Communications 29(2), 246 – 256 (2006), dependable Wireless Sensor Networks

11. Insfran, E., Pastor, O., Wieringa, R.: Requirements engineering-based conceptual modeling. Requirements Engineering Journal 7, 61–72 (2002)
12. Jamshidi, P., Khoshnevis, S., Teimourzadegan, R., Nikravesh, A., Shams, F.: Toward automatic transformation of enterprise business model to service model. In: PESOS '09: Proceedings of the 2009 ICSE Workshop on Principles of Engineering Service Oriented Systems. pp. 70–74. IEEE Computer Society, Washington, DC, USA (2009)
13. Kherraf, S., Lefebvre, E., Suryn, W.: Transformation from cim to pim using patterns and archetypes. Software Engineering, 2008. ASWEC 2008. 19th Australian Conference on pp. 338 –346 (march 2008)
14. Kitchenham, B.: Procedures for performing systematic reviews. Tech. rep., Keele University and NICTA (2004)
15. Koch, N., Zhang, G., Escalona, M.J.: Model transformations from requirements to web system design. In: ICWE '06: Proceedings of the 6th international conference on Web engineering. pp. 281–288. ACM, New York, NY, USA (2006)
16. Laguna, M.A., Gonzalez-Baixauli, B.: Requirements variability models: metamodel based transformations. In: MIS '05: Proceedings of the 2005 symposia on Metainformatics. p. 9. ACM, New York, NY, USA (2005)
17. Lapouchnian, A., Yu, Y., Liaskos, S., Mylopoulos, J.: Requirements-driven design of autonomic application software. In: CASCON '06: Proceedings of the 2006 conference of the Center for Advanced Studies on Collaborative research. p. 7. ACM, New York, NY, USA (2006)
18. Letier, E., van Lamsweerde, A.: Deriving operational software specifications from system goals. In: SIGSOFT '02/FSE-10: Proceedings of the 10th ACM SIGSOFT symposium on Foundations of software engineering. pp. 119–128. ACM, New York, NY, USA (2002)
19. Li, X., Liu, Z.: Prototyping system requirements model. Electronic Notes in Theoretical Computer Science 207, 17 – 32 (2008), proceedings of the 1st International Workshop on Harnessing Theories for Tool Support in Software (TTSS 2007)
20. Liew, P., Kontogiannis, K., Tong, T.: A framework for business model driven development. In: Software Technology and Engineering Practice, 2004. STEP 2004. The 12th International Workshop on. pp. 8 pp. –56 (sept 2004)
21. Machado, R., Fernandes, J., Monteiro, P., Rodrigues, H.: Transformation of uml models for service-oriented software architectures. In: ECBS '05: 12th IEEE International Conference and Workshops on the Engineering of Computer-Based Systems, 2005. pp. 173 – 182 (april 2005)
22. Marschall, F., Schoenmakers, M.: Towards model-based requirements engineering for web-enabled b2b applications. In: ECBS'03: 10th IEEE International Conference and Workshop on the Engineering of Computer-Based Systems. pp. 312–320. IEEE Computer Society, Huntsville, AL, EUA (April 2003)
23. Mauco, M., Leonard, M., Riesco, D., Montejano, G., Debnath, N.: Formalising a derivation strategy for formal specifications from natural language requirements models. Proceedings of the Fifth IEEE International Symposium on Signal Processing and Information Technology, 2005. pp. 646 –651 (dec 2005)
24. Mazón, J.N., Trujillo, J., Lechtenbrger, J.: Reconciling requirement-driven data warehouses with data sources via multidimensional normal forms. Data & Knowledge Engineering 63(3), 725 – 751 (2007)
25. Mendes, E.: A systematic review of Web engineering research. In: ISESE05: International Symposium on Empirical Software Engineering. pp. 498–507. IEEE (2005)

26. Mens, T., Czarnecki, K., Gorp, P.V.: 04101 discussion – a taxonomy of model transformations. In: Bezivin, J., Heckel, R. (eds.) Language Engineering for Model-Driven Software Development. No. 04101 in Dagstuhl Seminar Proceedings, Internationales Begegnungs- und Forschungszentrum für Informatik (IBFI), Schloss Dagstuhl, Germany, Dagstuhl, Germany (2005)
27. Menzies, T.: Editorial: model-based requirements engineering. Requir. Eng. 8(4), 193–194 (2003)
28. Naslavsky, L., Alspaugh, T.A., Richardson, D.J., Ziv, H.: Using scenarios to support traceability. In: TEFSE '05: Proceedings of the 3rd international workshop on Traceability in emerging forms of software engineering. pp. 25–30. ACM, New York, NY, USA (2005)
29. Nicolás, J., Toval, A.: On the generation of requirements specifications from software engineering models: A systematic literature review. Inf. Softw. Technol. 51(9), 1291–1307 (2009)
30. Ozkaya, I., Akin, O.: Requirement-driven design: assistance for information traceability in design computing. Design Studies 27(3), 381 – 398 (2006)
31. Petticrew, M., Roberts, H.: Systematic Reviews in the Social Sciences: A Practical Guide. Blackwell Publishing (2006)
32. Raj, A., Prabhakar, T.V., Hendryx, S.: Transformation of sbvr business design to uml models. In: ISEC '08: Proceedings of the 1st conference on India software engineering conference. pp. 29–38. ACM, New York, NY, USA (2008)
33. Rash, J., Hinchey, M., Rouff, C.: Formal requirements-based programming for complex systems. In: ICECCS 2005: Proceedings of the 10th IEEE International Conference on Engineering of Complex Computer Systems, 2005. pp. 116 – 125 (june 2005)
34. Regnell, B., Runeson, P., Wohlin, C.: Towards integration of use case modelling and usage-based testing. Journal of Systems and Software 50(2), 117 – 130 (2000)
35. Santos, J., Moreira, A., Araujo, J., Amaral, V., Alferez, M., Kulesza, U.: Generating requirements analysis models from textual requirements. In: MARK '08: First International Workshop on Managing Requirements Knowledge, 2008. pp. 32 –41 (sept 2008)
36. Shin, J.E., Sutcliffe, A.G., Gregoriades, A.: Scenario advisor tool for requirements engineering. Requirements Engineering 10, 2005 (2004)
37. Sousa, K., Mendonça, H., Vanderdonckt, J., Pimenta, M.S.: Supporting requirements in a traceability approach between business process and user interfaces. In: IHC '08: Proceedings of the VIII Brazilian Symposium on Human Factors in Computing Systems. pp. 272–275. Sociedade Brasileira de Computaç ao, Porto Alegre, Brazil, Brazil (2008)
38. Zhang, L., Jiang, W.: Transforming business requirements into bpel: A mda-based approach to web application development. In: WSCS '08: IEEE International Workshop on Semantic Computing and Systems, 2008. pp. 61 –66 (july 2008)