MODEL-DRIVEN ENGINEERING AS A NEW LANDSCAPE FOR TRACEABILITY MANAGEMENT: A SYSTEMATIC LITERATURE REVIEW

Iván Santiago*, Álvaro Jiménez, Juan Manuel Vara, Valeria De Castro, Verónica A. Bollati, Esperanza Marcos

Kybele Research Group, Department of Computing Languages and Systems II, Rey Juan Carlos University, 28933, Móstoles, Madrid, Spain.

E-mail addresses: {ivan.santiago, alvaro.jimenez, juanmanuel.vara, valeria.decastro, veronica.bollati, esperanza.marcos}@urjc.es

Abstract. Context: Model-Driven Engineering provides a new landscape for dealing with traceability in software development. Objective: Our goal is to analyze the current state of the art in traceability management in the context of Model-Driven Engineering. Method: We use the systematic literature review based on the guidelines proposed by Kitchenham. We propose five research questions and seven quality assessments. Results: Of the 157 relevant studies identified, 29 have been considered primary studies. These studies have resulted in 17 proposals. Conclusion: The quality evaluation shows that the most addressed operations are storage, CRUD and visualization, while the most immature operations are exchange and analysis traceability information.

Keywords. *Traceability, Model-Driven Engineering, Systematic Literature Review.*

1. Introduction

The IEEE [1] (pp. 78) defines traceability as: "*the degree to which a relationship can be established between two or more products of the development process, especially products having a predecessor-successor or master-subordinate relationship to one another; for example, the degree to which the requirements and design of a given software component match*". Traceability implies keeping track of the relationships between requirements, design artifacts, source code, test cases, etc. and it has always been a relevant topic in Software Engineering (SE) [2]. The evolution of system components throughout the development process can be monitored by means of the appropriate management of traceability information. It also allows us to establish relationships between elements specified as requirements and elements implemented in the final system [3].

In addition, the information obtained from traceability management can be used in different activities, such as change impact assessment, regression testing, requirements validation, code coverage analysis, etc [4-8]. Furthermore, traceability information can be used to make the

* *Corresponding author:* Iván Santiago (*ivan.santiago@urjc.es*). Tel.: +34 91 4889359; fax: +34 91 4888530

software more understandable [9-11], to support design decisions [12, 13], to improve configuration management [14], for product line development [14, 15] and obviously, to assist in maintenance tasks [16].

Unfortunately, according to Oliveto [17], Egyed [18], Hayes et al. [19] and Mäder et al. [20], the lack of automatic or semi-automatic support hampers the maintenance of links among software artifacts, resulting in a tedious and time consuming task. Probably for this reason, traceability information becomes obsolete very quickly during software development and sometimes it is completely omitted.

However, the advent of Model-Driven Engineering (MDE) [21], which principles are to enhance the role of models and modeling activities and to increase the level of automation all along the development process [22], can drastically change this landscape. The key role of models in any MDE development process can decisively help to facilitate trace maintenance. The software assets handled in an MDE software development process are mainly models; regardless of whether they are represented graphically or textually. Therefore, trace maintenance can be seen mainly as links between the elements of those models. Furthermore, the traces could be collected in other models and, therefore, processed using any model processing technique, such as model transformation, model matching or model merging [23], etc. Moreover, if the models considered in the development process are connected by a model transformation and the language used to develop the transformation provides support to keep the trace information, such information can be generated automatically [24]. Thus, if an element from a source model is modified, this modification could be propagated to the corresponding elements in the target model. This scenario is represented by a very simplistic example in Figure 1: two given models (Ma and Mb) are connected by a model transformation (MMa2MMb). The transformation maps squares and circles from the source model (Ma) into cubes and cylinders in the target model (Mb). To keep track of these relationships after the transformation has been executed, it would be desirable to have at one's disposal an "extra" model of trace objects (MTrace).

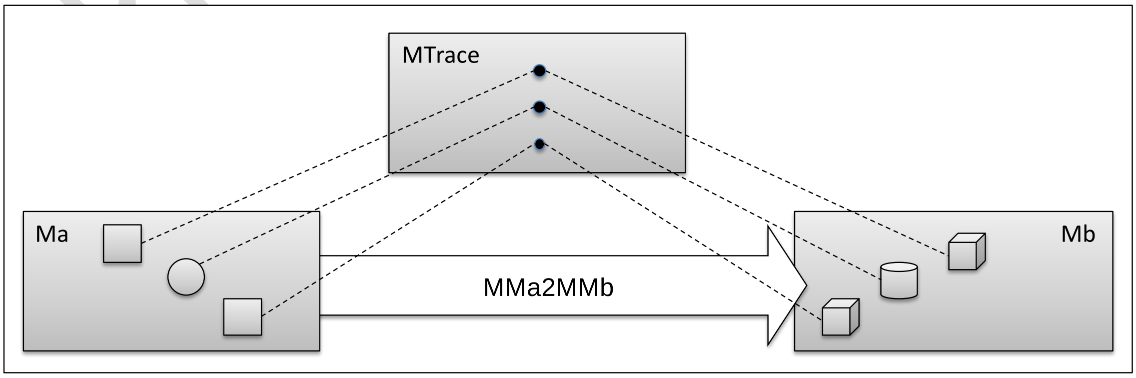

Figure 1. Example of the trace links collected in a MDE scenario

Given the relevance of traceability management in software development and the appearance of a new scenario that offers the possibility to make improvements to the management of traceability in traditional software engineering, we consider it of great interest to perform a review of the literature regarding this subject. Our goal is to discover how traceability is managed and analyzed in the context of MDE and to identify possible improvements. In order to provide a comprehensive review of the topic, this work presents a Systematic Literature Review [25-28] of traceability management in the context of MDE that answers research questions such as: What level of automation is recommended for the generation of trace links? or What are the limitations of the current state-of-the-art in traceability management in the context of MDE?

As the IEEE definition has shown, the term *trace* conveys some special usages (*predecessor-successor or master-subordinate* relationships). However, in this work we are interested in looking at traceability in a more generic manner. Thus, before exploring the contents of this study, we would like to provide precise definitions of some terms used throughout the paper. In the context of this work, a **traceability relationship** is a correspondence between two or more types of elements, whereas a **trace-link** (or simply a trace) is an instance of this relationship. These relationships encode different semantics, such as provenance or the aforementioned master-subordinate [29]. In particular, trace-links are frequently used in an MDE scenario to encode the relationships between source and target objects of a model transformation whereas some form of weaving model is used to collect such traces [23]. Finally, t**raceability information** is generally obtained from one or more trace links, i.e. trace links are the raw material for the construction of traceability information.

The remainder of this paper is structured as follows: we will describe the method we followed in Section 2 and present our results in Section 3. In Section 4 we will answer our major research questions and in Section 5 compare our work with related works. Finally, in Section 6, we will present our conclusions.

2. Method

The research method used in this study is a systematic literature review based on the guidelines proposed by Kitchenham [28] and Biolchini et al. [26]. Figure 2 shows an overview of the process. According to these guidelines, a systematic literature review process is composed of three consecutive stages: *planning*, *execution* and *result analysis*; and another stage which is performed throughout the whole process in order to store the results of the previous stages: *packaging*. Thus there are two checkpoints in the course of the process to evaluate that the systematic literature review process executed is correct [26].

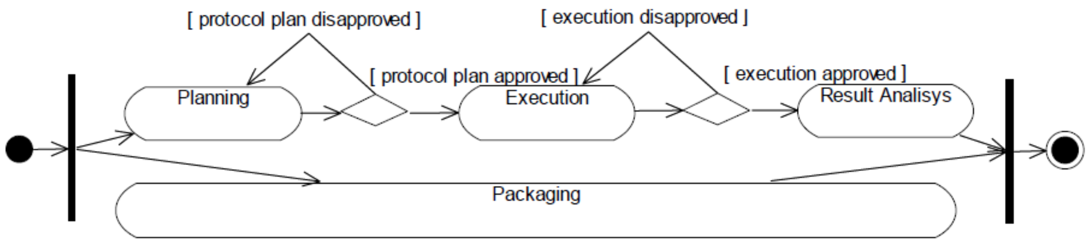

Figure 2. Systematic literature review process proposed by Biolchini et al. in [26]

In this section, we will focus on the planning stage which involves defining the research objectives and the way the review was carried out. Concretely in the following subsections, we will present: research questions (section 2.1), the data sources and query string we used to execute searches (section 2.2), the inclusion and exclusion criteria (section 2.3), the quality assessment (section 2.4), the data extracted from each selected study (section 2.5) and the process used to conduct the review (section 2.6).

2.1. Research questions

The first step we carried out in the planning stage was to define the principal goal of the systematic literature review, which is: to identify and analyze the state of the art in traceability management in the context of the Model-Driven Engineering. In order to achieve this goal, we defined a set of research questions (RQ) to be addressed by this review:

- **RQ1**. What level of automation is suggested by methodological proposals for the generation of trace links?

  As we mentioned before, the level of automation is key to getting a full return on MDE promises of faster, less costly software development [22]. Therefore, we are interested in the level of automation of the trace links generation proposed. Some proposals suggest a methodology which requires users to make a considerable effort during the generation process, e.g. defining relationships or refining the results [30]; other proposals provide techniques to raise the automation level and alleviate these efforts, e.g. taking implicit traceability relationships from other model management tasks such as model transformations [31].
- **RQ2**. How do methodological proposals suggest that traceability should be managed and analyzed?

  As mentioned in the introduction, trace links can be used to carry out different tasks during the software development process (e.g. changing impact analysis, improving the configuration management, system maintenance, etc.) [9-16]. It is desirable to know the way in which the different proposals deal with certain tasks related to the management of traceability information, such as storage (e.g. in models, in repositories, embedded in other artifacts, etc.), representation (textual or graphical) or analysis techniques (e.g. reports, statistic information, classifications, etc.). Furthermore, we are interested in discovering which operations traceability management implies according to each proposal. This might typically be creation, updating, deletion, querying, exchange, etc.

- **RQ3**. Are there tools or frameworks that provide technological support for the management of traceability in the context of MDE?
  One of the main goals of this review is to find out whether traceability management has been put into practice by means of MDE tools. Therefore, the aim of this research question is to analyze whether each proposal provides the user with some kind of technological support.
- **RQ4**. What are the limitations of state of the art in traceability management in the context of MDE?
  Finally, the aim of this question is to combine the answers to the previous research questions (RQ1, RQ2 and RQ3) in order to identify gaps or limitations when dealing with traceability management in MDE proposals. The goal is to evaluate whether there is space for improvement in the area and to assess the possibility of addressing them in future works.
- **RQ5**. Are there forums (e.g. journals or conferences) that specialize in dealing with traceability management in MDE?
  We identified where studies related to traceability management in MDE have been published in order to analyze whether there are any journals or conferences that specialize in this topic and to know how many forums have accepted studies in this area. Moreover, we wanted to know how relevant those forums were according to accepted classifications, such as JCR or CORE rankings.

2.2. Data sources and query strings

The planning stage also involves enumerating data sources which we will search for studies or previous works and will use to define the query strings that would be executed on those sources [26, 28]. The following digital libraries were selected to carry out the search process of this review (*Name [Acronym]: website*):

- ACM Digital Library [ACM]: http://portal.acm.org/
- CiteSeerX [CSX]: http://citeseerx.ist.psu.edu/
- IEEEXplore [IEEEX]: http://ieeexplore.ieee.org/
- Google Scholar [GS]: http://scholar.google.com/
- ISI Web of Knowledge [ISI]: http://www.webofknowledge.com/
- Science Direct [SD]: http://www.sciencedirect.com/
- SpringerLink [SL]: http://www.springerlink.com/
- The Collection of Computer Science Bibliographies [CSB]:

http://liinwww.ira.uka.de/bibliography/

Since each of these digital libraries use its own syntax, the *canonical* query string defined for this review has been adapted to each search engine, as shown in Table 1. All in all, we combined some words related to the main goal of this study with some logical operators ("OR", "AND") to define the following query string: *traceability AND (mda OR mdsd OR mdd OR mde OR model driven).*

2.3. Inclusion and exclusion criteria

Although the query string is defined by taking the main goal of the review in to consideration, we should consider the possibility that a certain number of studies obtained from the searches do not provide any evidence related to the research questions posed. According to [28], it is necessary to define inclusion and exclusion criteria based on the research questions defined in order to filter those studies.

In this review, we included studies published online before March 2011 fulfilling at least one of the following criteria:

- Its abstract led us to conclude that the main purpose of the study is traceability management in MDE
- Its title or keywords included the strings: "traceability" and "model driven".

Actually, studies that fulfill the inclusion criteria exist but they do not provide relevant information regarding the main goal of this review. Therefore, we had to read each study in detail keeping in mind the following exclusion criteria:

- Studies that acknowledge traceability management as a desirable task but do not provide the technique to put it into practice.
- Studies dealing with traceability without considering MDE assets, i.e. models and/or model transformations.
- Studies whose main purpose is to classify other articles or are systematic literature reviews themselves (these articles are considered as secondary studies). These studies will be compared with our review in section 5.

2.4. Quality Assessment

Once we have selected a number of works that are in agreement with the inclusion and exclusion criteria we should be able to assess the quality of the research they present. Therefore, and according to the guidelines proposed by Kitchenham [28], six Quality Assessment (QA) questions have been defined in order to assess the quality of the research of each proposal and to provide a quantitative comparison between them. The scoring procedure used was Yes (Y) = 1, Partly (P) = 0.5 or No (N) = 0. The quality assessment questions defined in this systematic literature review were:

- **QA1.** How clear and coherent is the work?
- **QA2**. How clear is the research goal defined?
- **QA3.** How well can the route from the research goal to any conclusion be seen?
- **QA4**. How clearly is the research process established?
- **QA5**. How good is the work in comparison to other related works?
- **QA6**. How clearly are the work limitations documented?

These questions can help us to check the biases, the external validation and the internal validation of the proposals.

2.5. Data extraction and analysis

The data extraction phase allows for the gathering of all the information needed from the different studies selected in order to be able to answer the research questions [28].

In this review, we first collected some basic data to identify each study such as:

- Title and authors.
- Abstract.
- Publication (e.g. journal or conference proceedings) and publication year.

Furthermore, in order to be able to carry out an in-depth analysis of the proposals, we decided to extract the following information from each of the primary studies (those that fulfilled the inclusion criteria and were not excluded by the exclusion criteria):

- Whether traceability relationships are defined automatically (e.g. from transformations) or manually by users.
- The type of transformations used to manage traceability information (model-to-model or model-to-text transformations).
- The transformation languages used to manage traceability information.
- The form in which traceability links should be saved according to the proposal, e.g. in trace models, in a trace repository, embedded in the very same models handled during the development process, i.e. there is no specific container for traceability information.
- The type of traceability metamodel proposed or used (generic or specific), if applicable. A generic metamodel does not refer to a universal metamodel, but to a metamodel that can be used in every traceability scenario. That is, each proposal could have its own generic metamodel. On the other hand, a specific or ad-hoc metamodel refers to a metamodel devised for dealing with a particular scenario for traceability management.
- Modeling languages from which traces can be derived (e.g. UML, EMF)
- The way in which traceability information is visualized and/or represented.
- The operations considered for the management of traceability information, e.g. creation, query, updating, deletion, etc.
- The analysis of traceability information performed to carry out other tasks.

- Whether the proposal has been implemented, i.e. to find out if there is any tool that supports the theoretical proposal.

In order to simplify the analysis of these data, the primary studies were put into small Groups of Primary Studies (GPS). Each GPS contained those studies which have one or more authors in common and their ideas belong to the same or a very similar line of research, so that they are mainly an evolution of the same initial hypothesis. That is, each GPS can be seen as the set of interrelated publications related to a given proposal.

Using this grouping, the data extracted from each GPS and the quality assessment values for each of them, we are in a position to answer the research questions RQ1-RQ4. Furthermore, the basic data extracted allows for the categorizing of the studies according to its publication, which helps to answer RQ5 (to identify forums devoted to traceability in MDE).

2.5.1. GPSs assessment

In order to improve the analysis of GPSs with regard to the context of the research area, we defined seven GPS assessment (GA) questions and scores for them. The scoring procedure used was Yes (Y) = 1, Partly (P) = 0.5, No (N) = 0, or Unknown (i.e. the information is not specified). The GPSs assessment questions defined in this systematic literature review were:

- **GA1**. Does it suggest or consider any techniques to improve the level of automation for the generation of trace links?

  Scores: Y (Yes), the methodological proposal does provide ideas or techniques about how the generation of traces should be automated; P (Partly), it suggests techniques to generate traces semi-automatically; N (No), the proposal does not consider automating the generation of traces.
- **GA2**. Does it suggest a technique for the storage of trace links?

  Scores: Y, it suggests mechanisms to materialize trace links in some form of data structure. Two options are mainly considered: external traceability, required from database schemes or ad-hoc metamodels and internal traceability, where trace links are collected in the models themselves by means of textual references or hyperlinks; P, it considers the storage of trace links or acknowledges that it has to be supported but does not provide any mechanism to do so; N, the proposal does not consider the storage of trace links.
- **GA3**. Does it suggest a mechanism for visualizing (representing) trace links?

  Scores: Y, it suggests the development of an ad-hoc tool to support trace links visualization or some other graphical representation, like a traceability matrix; P, it leans on generic tools to display trace links, instead of using tools devised to represent this kind of information. For instance, if trace links are stored in a model, they can be displayed by a generic model editor, such as EMF generated tree-like editor [32], and the resulting representation cannot be optimal or intuitive; N, the proposal does not suggest any form of representing trace links.

- **GA4**. Does it consider the creation, deletion, modification and/or querying of trace links? And if so, does it states the way in which such tasks should be supported?
  Scores: Y, the proposal considers the four tasks; P, the proposal does not consider all of these tasks, only some of them; N, the proposal does not consider any of them.
- **GA5**. Does it provide ideas or techniques for the interchange[1] of traceability information between different proposals?
  Scores: Y, the proposal addresses the problem of traceability information interchange offering solutions; P, the proposal only identifies the interchange problem, but it does not offer any solution; N, the proposal does not consider the interchange of traceability information.
- **GA6**. Does it perform some type of analysis with the traceability information obtained?
  Scores: Y, the proposal aims to analyze traceability information with the goal of providing refined information to the different stakeholders; P, the proposal considers interesting to classify or analyze the information obtained from traces, but it does not state the way to do so; N, the proposal does not consider the analysis of traceability information.
- **GA7**. Does it provide any implementation, tool or technological support?
  Scores: Y, it includes a complete framework or tool to support its methodological proposal; P, it provides us with a partial implementation of the proposal; N, it is only a theoretical proposal, lacking implementation.

2.6. Process to conduct the review

The last step in the planning stage is to define the process to conduct the review. In this case, we defined a process based on the one proposed by Pino et al. in [27]. It basically consists of the three phases shown in Figure 3: search process (Figure 3.a), primary studies selection (Figure 3.b) and data extraction (Figure 3.c).

[1] Adopting the terminology used by the Model Interchange Working Group (http://www.omg.org/news/releases/pr2009/07-08-09.htm), we use the term **interchange** to refer globally to the need of import/export mechanisms to exchange information between different proposals.

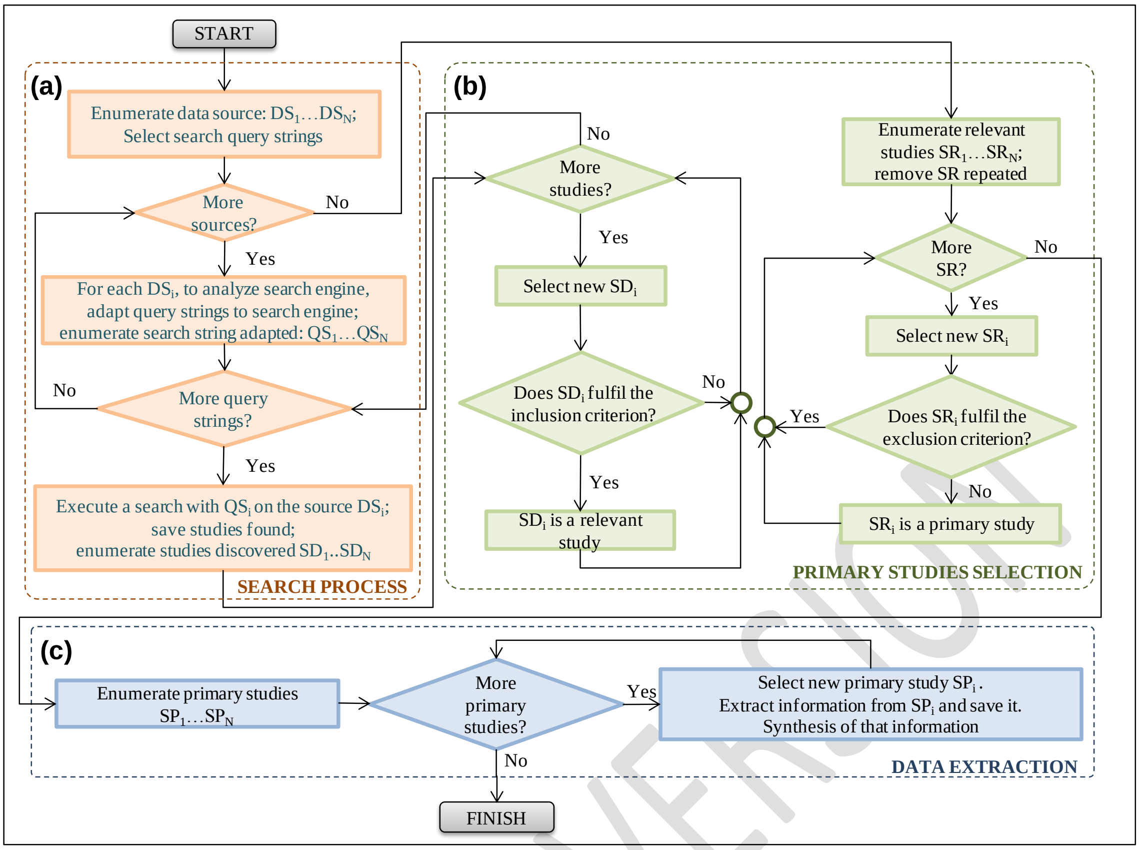

Figure 3. Process followed to conduct the review: (a) search process, (b) primary studies selection and (c) data extraction

The first step in the search process consists of enumerating the data sources and the query strings used to search for studies. Once the data sources have been identified, we have to deal with their search engines. For this purpose, the search engine of each data source (DS_i) is analyzed in order to adapt the query strings to the syntax of each search engine. Then, each adapted query string (QS_i) is used to search for studies in the corresponding data source (DS_i). As mentioned before, in this study we have defined one query string (see section 2.2) which was adapted to each search engine chosen as Table 1 shows. Additionally, it shows the search scope for each digital library.

Table 1. Adapted Query Strings

Digital Library	Adapted Query String	Scope
ACM	(traceability) and (mda or mdsd or mdd or mde or "model driven")	Title, Abstract, Review
CSX	((traceability) AND (mda OR mdsd OR mdd OR mde OR "model driven"))	Text
IEEEX	(traceability AND (mda OR mdsd OR mdd OR mde OR "model driven"))	All
GS	traceability + (mda OR mdsd OR mdd OR mde OR "model driven")	All
ISI	TS=(traceability AND (mda OR mdsd OR mdd OR mde OR "model driven"))	Topic
SD	ALL((traceability) AND (mda OR mdsd OR mdd OR mde OR "model driven"))	All

Digital Library	Adapted Query String	Scope
SL	(traceability AND (mda OR mdsd OR mdd OR mde OR "model driven"))	All
CSB	traceability +(mda mdsd mdd mde "model driven")	Author, Title

The next step consists of storing and enumerating all the studies returned by each data source. As mentioned in the previous section, this step implies extracting some data (title, authors, abstract, publication and year of online publication) from each study.

The Primary Studies selection process is then tackled (Figure 3.b). This implies checking whether each of the studies returned (SD_i) fulfills the inclusion criteria defined in Section 2.3. That is, whether it was published online before March 2011, its title or keywords include the strings: “traceability” and “model driven” or its abstract provides us with sufficient evidence to state that its main focus is traceability management in MDE. If the study fulfills the inclusion criteria, it becomes a relevant study in the systematic literature review.

After having identified all the relevant studies ($SR_1 \ldots SR_N$) recovered from the different data sources, it is time to address the second stage of the Primary Studies selection process. Basically, it consisted of two tasks: removing possible duplicates that could appear due to the same search being executed in different search engines; in addition, each non-duplicate study is evaluated according to the exclusion criteria defined in Section 2.3 to identify those studies that must be excluded from the final set of primary studies that are used to perform the last phase of the process: data extraction (Figure 3.c). In this last phase, each primary study (SP_i) is analyzed in detail to extract the information defined in Section 2.5. Moreover, we analyze the bibliographical references cited by each primary study in order to seek other works that may be relevant to the research purpose of this review.

3. Results

This section presents the main results obtained from this systematic literature review performed according to the method described in the previous section. Note that the analysis of primary studies was performed using the different Groups of Primary Studies (GPS) as the information unit. Accordingly, the results that will be presented in this section correspond to those GPS. For instance, the different Quality Assessment questions were not answered for each primary study but each Group of Primary Studies.

3.1. Search and Primary Studies selection

As mentioned previously when describing the process followed to conduct the review, the first step was to execute the search in each data source with the corresponding query string adapted to the syntax of the search engine. Additionally, to maximize the number of results recovered, we executed the search using the widest scope allowed by the search engine. Table 1 summarizes the query string and the scope for each data source. Executing these searches, we obtained a total of 10,028 results (note: Google scholar only returns the first 1,000 results). The first two columns on the left of Table 2 show the number of results for each data source.

The next stage of the method described in Figure 3 was the selection of primary studies, i.e. to evaluate if each study recovered fulfilled the inclusion criteria. In this way, only 267 out of the 10,028 recovered studies were found to be relevant, i.e. 2.66 per cent. Again, these results are broken down in column 3 of Table 2 to show the number of relevant studies found in each digital library, whereas column 4 shows the percentage of relevant studies returned by each source, e.g. 9.71% per cent of the studies returned by the ACM digital library were identified as relevant studies. Accordingly, it is worth noting that The Collection of Computer Science Bibliographies (CSB) and ISI Web of Knowledge (ISI) garnered the highest percentages (31.48% and 29.07%, respectively). Indeed, they are also the two search engines that returned the smallest number of results (CSB: 54 and ISI: 86). Therefore, we consider them to be the most accurate search engines in terms of the number of relevant studies returned. With regards to the low number of relevant studies identified out of the total number of studies returned by the different search engines (2.66%), this is mainly due to the fact that many of the returned studies were found to contain some words of the query string, but when analyzed, it turned out that they were not related to the research questions of this systematic literature review, so they were excluded.

In turn, column 5 shows the percentage of relevant studies found in each digital library with regards to the total number of relevant studies identified, e.g. ACM returned 40 relevant studies, which constitutes the 14.98% of the 267 relevant studies found in this review. From the data in this column we can see that most of the relevant studies were found in Google Scholar (30.71%), SpringerLink (17.60%) and ACM (14.98%). However, this data is not very conclusive since many of those relevant studies were also found in other data sources. For instance, about 75% of the relevant studies found in Google Scholar can also be found in one of the other digital libraries. These repeated studies correspond to publications in high impact forums such as CORE-A conferences or JCR journals.

Table 2. Search results

Digital Library (DL)	Search Results	Relevant Studies	% of Relevant studies	% of ALL the Relevant studies
ACM	412	40	9.71 %	14.98 %
CSX	451	26	5.76 %	9.74 %
IEEEX	923	21	2.28 %	7.87 %
GS	6980[2]	82	1.17 %	30.71 %
ISI	86	25	29.07 %	9.36 %
SD	259	9	3.47 %	3.37 %
SL	863	47	5.45 %	17.60 %
CSB	54	17	31.48 %	6.37 %
All Libraries	**10028**	**267**	**2.66 %**	**100 %**

Table 3 addresses the matter of duplicates by showing that more than 40% of the relevant studies found in this review were duplicated in different data sources. Therefore, we removed those copies and consequently we obtained a list of 157 non-duplicated relevant studies. Since a given study could be found in several data sources and there is no criteria to select it from one of specific data source as we do not classify the studies according to the digital library where they were found.

Table 3. Filter of relevant studies

	#Studies	Percentage
Relevant studies	267	100 %
Duplicated relevant studies	110	41.20 %
Non-duplicated relevant studies	157	58.80 %

Next, each non-duplicated relevant study was evaluated according to the exclusion criteria. As a result, only 29 of them (i.e. 18.47% of the non-duplicated relevant studies) became the primary studies for this systematic literature review. Table 4 contains the complete list of primary studies grouped into Groups of Primary Studies according to their commonalities in terms of authors and proposed ideas, as we have explained in section 2.5.

It is worth mentioning that we have analyzed the references of these primary studies in order to identify other relevant works. However, we have not found any relevant works that are different from those already detected.

Table 4. Primary Studies

ID	Authors	Title	Year	Publication
GPS1	Grammel and Kastenholz	A generic traceability framework for facet-based traceability data extraction in model-driven software development [33]	2010	ECMFA

[2] Google scholar returned 6980 results, but it only displays the first 1000 of them.

ID	Authors	Title	Year	Publication
GPS2	Anquetil, Kulesza, Mitschke, Moreira, Royer, Rummler and Sousa	A model-driven traceability framework for software product lines [14]	2010	SOSYM
	Sousa, Kulesza, Rummler, Anquetil, Mitschke, Moreira, Amaral and Araujo	A model-driven traceability framework to software product line development [15]	2008	ECMFA
GPS3	Aleksy, Hildenbrand, Obergfell and Schwind,	A Pragmatic Approach to Traceability in Model-Driven Development [34]	2009	PRIMIUM
GPS4	Drivalos, Kolovos, Paige and Fernandes	A state-based approach to traceability maintenance [35]	2010	ECMFA
	Drivalos, Kolovos, Paige and Fernandes	Engineering a DSL for software traceability [36]	2009	SLE
	Kolovos, Paige and Polack	On-Demand Merging of Traceability Links with Models [37]	2006	ECMFA
	Paige, Drivalos, Kolovos, Fernandes, Power, Olsen and Zschaler	Rigorous identification and encoding of trace-links in model-driven engineering [38]	2011	SOSYM
	Drivalos, Paige, Fernandes and Kolovos	Towards Rigorously Defined Model- to-Model Traceability [39]	2008	ECMFA
GPS5	Levendovszky, Balasubramanian, Smyth, Shi and Karsai	A transformation instance-based approach to traceability [40]	2010	ECMFA
GPS6	Kurtev, Van den Berg and Jouault	Evaluation of rule-based modularization in model transformation languages illustrated with ATL [41]	2006	SAC
	Jouault, Vanhooff, Bruneliere, Doux, Berbers and Bézivin	Inter-DSL coordination support by combining megamodeling and model weaving [42]	2010	SAC
	Jouault	Loosely Coupled Traceability for ATL [31]	2005	ECMFA
	Jossic, Del Fabro, Lerat, Bézivin and Jouault	Model integration with model weaving: A case study in system architecture [43]	2007	ICSEM
	Allilaire	Towards traceability support in ATL with Obeo Traceability [44]	2009	MT-ATL
	Barbero, Del Fabro and Bézivin	Traceability and provenance issues in global model management [45]	2007	ECMFA
GPS7	Boronat, Carsí and Ramos	Automatic support for traceability in a generic model management framework [46]	2005	ECMFA
GPS8	von Pilgrim, Vanhooff, Schulz-Gerlach and Berbers	Constructing and Visualizing Transformation Chains [47]	2008	ECMFA
GPS9	Guerra, de Lara, Kolovos and Paige	Inter-modelling: From theory to practice [48]	2010	MODELS
GPS10	Valderas and Pelechano	Introducing requirements traceability support in model-driven development of web applications [5]	2009	INFSOF
GPS11	Sánchez, Alonso, Rosique, Álvarez and Pastor	Introducing safety requirements traceability support in model-driven development of robotic applications [49]	2011	IEEE-TC
GPS12	Yu, Lin, Hu, Hidaka and Kato	Maintaining invariant traceability through bidirectional transformations for EMF [50]	2010	TR- GRACE
GPS13	Olsen and Oldevik	Scenarios of Traceability in Model to Text Transformations [29]	2007	ECMFA
	Melby	Traceability in Model Driven Engineering [51]	2007	UO
	Oldevik and Neple	Traceability in Model to Text Transformations [52]	2006	ECMFA
GPS14	Vanhooff and Berbers	Supporting modular transformations units with precise transformation traceability metadata [53]	2005	ECMFA
GPS15	Walderhaug, Johansen, Stav and Aagedal	Towards a Generic Solution for Traceability in MDD [54]	2006	ECMFA
GPS16	Falleri, Huchard and Nebut	Towards a traceability framework for model transformations in kermeta [30]	2006	ECMFA
GPS17	Bonde, Boulet and Dekeyser	Traceability and interoperability at different levels of abstraction in model-driven engineering [55]	2006	ASDLS

3.2. Quality assessment results

Once the primary studies of the Systematic Literature Review had been identified, we evaluated them according to the quality assessment questions defined in section 2.4. The score assigned to each study for each question is shown in Table 5.

The results show that all of the GPSs obtained some points: while only three of them scored less than or equal to 3 points, eight reached 4.5 or more points (the statistical mode). The last column ("% max GPS") shows the percentage attained by each GPS out of the total score (i.e., 6). The penultimate row ("% total score") shows the percentage of points obtained by all the GPS with regard to the total number of points obtained by all the GPSs in all the Quality Assessment questions. Finally, the last row ("% max GA") corresponds to the percentage of points collected by the values assigned to a given Quality Assessment question out of the points that would be collected if every GPS obtained the highest score (i.e. 17 * 1 = 17). The arithmetic mean of the scores is 4.03 and the standard deviation is 0.61.

Table 5. Quality assessment of GPSs

ID	QA1	QA2	QA3	QA4	QA5	QA6	Total Score	% by Max GPS
GPS1	Y	Y	Y	N	P	P	**4**	**66.67%**
GPS2	Y	Y	P	N	Y	Y	**4.5**	**75.00%**
GPS3	Y	P	P	P	Y	Y	**4.5**	**75.00%**
GPS4	Y	Y	Y	N	Y	Y	**5**	**83.33%**
GPS5	Y	P	P	N	Y	Y	**4**	**66.67%**
GPS6	Y	Y	Y	N	P	Y	**4.5**	**75.00%**
GPS7	Y	Y	P	N	Y	N	**3.5**	**58.33%**
GPS8	Y	Y	Y	N	Y	P	**4.5**	**75.00%**
GPS9	Y	Y	Y	N	Y	P	**4.5**	**75.00%**
GPS10	Y	Y	Y	N	Y	N	**4**	**66.67%**
GPS11	Y	Y	Y	N	Y	P	**4.5**	**75.00%**
GPS12	Y	Y	Y	N	N	N	**3**	**50.00%**
GPS13	Y	Y	Y	N	Y	P	**4.5**	**75.00%**
GPS14	Y	Y	P	N	Y	P	**4**	**66.67%**
GPS15	Y	Y	P	N	P	N	**3**	**50.00%**
GPS16	Y	Y	P	N	N	Y	**3.5**	**58.33%**
GPS17	Y	Y	Y	N	N	N	**3**	**50.00%**
Total	**17**	**16**	**13.5**	**0.5**	**12.5**	**9**	**68.5**	
% total score	**24.82%**	**23.36%**	**19.71%**	**0.73%**	**18.25%**	**13.14%**	**100%**	
% by Max QA	**100%**	**94.12%**	**79.41%**	**2.94%**	**73.53%**	**52.94%**		

Figure 4 shows a pie chart depicting the distribution of the scores for the Quality Assessment questions. It illustrates that the first four questions are distributed out of 70% (67.88%) of the total score, while question 4 (research methodology) represent less than 1% (0.73%).

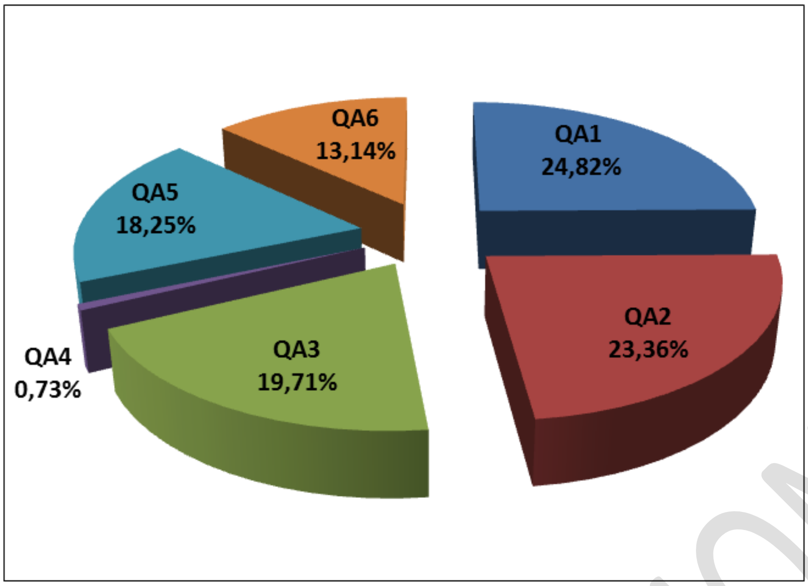

Figure 4. Score for Quality Assessment Questions

In view of these results, we can conclude that in general the quality of the research presented by the proposals evaluated is good since all of them obtain a minimum quality score of 50%. The best proposal according to the quality assessment performed is GPS4. This work obtains a positive score in all the areas assessed, with the exception of the specification of the research method followed.

Indeed, we can state that the specification of the research method is a major handicap of the works evaluated, since it is only partially addressed by only one proposal (GPS3), which references another work to indicate the research method followed. The other quality aspect that should be improved is that of documenting the limitations of the work, which is only addressed by 50% of the works reviewed. There is therefore, in some respects, a gap as regards the internal validation of the proposals.

3.3. Data extraction results

After identifying and grouping the primary studies we extracted the information specified in section 2.5 from each study. The data extracted is shown in part in Tables 5 and 6.

Table 6. Data extraction (Part I)

GPS	Relationships are defined ... (Auto/Manual)	Transformations used	Transformation languages used	Links are saved in…	Traceability metamodel
GPS1	Depends on the transformation engine which is extended	M2M and M2T	Depends on the transformation engine which is extended	Trace repository	Generic
GPS2	Auto or Manual	Not defined	Not defined	Trace repository	Generic
GPS3	Implicit links: Automatic Explicit links:	M2M and M2T	Velocity and XML serialization	XML/XMI models	Not provided

GPS	Relationships are defined ... (Auto/Manual)	Transformations used	Transformation languages used	Links are saved in…	Traceability metamodel
	Manual				
GPS4	Manual	M2M and M2T	Not defined	Trace models	Specific for each scenario
GPS5	Manual	M2M and M2T	GReAT and imperative languages	Graphs or text	Generic
GPS6	Auto or Manual	M2M	ATL	Trace models	Generic
GPS7	Automatic	Not defined	Not defined	Trace models	Generic
GPS8	Automatic	M2M	Java, ATL, MTF	Trace models	Generic
GPS9	Automatic	M2M	EOL	Trace models	Generic
GPS10	Manual	M2M	AGG	Navigational models	Generic
GPS11	Manual	M2M and M2T	ATL and JET	Trace models	Generic
GPS12	Manual	M2M	GRoundTram	Not generated	Not provided
GPS13	Auto and Manual	M2T	MOF Model to Text Transformation and MOFScript	Trace models	Generic
GPS14	Auto and Manual	M2M	Not defined	Embedded	Generic (UML profile)
GPS15	Manual	Not defined	Not defined	Trace repository	Generic
GPS16	Manual	M2M	Kermeta	Trace models	Generic
GPS17	Automatic	M2M	ModTransf (XML)	Trace models	Generic

Table 7. Data extraction (Part II)

GPS	Model languages supported	Information is visualized by…	Operations supported	Analysis of traces	Has been implemented?
GPS1	Trace-DSL	Not defined	Creation (C), updating (U), deletion (D) and query (Q)	Not provided	Yes
GPS2	Models produced by some tools (e.g. Rational Rose, MoPLine, Enterprise Architecture, etc.)	Graphs or textual representation	Exportation (E), importation (I), C, U, D and Q	Impact analysis query	Yes
GPS3	XML (XStream) and XMI (Ecore)	Graphical models (TraVis)	C and Q	Not provided	Yes
GPS4	XMI (Ecore)	Graphical models (ModeLink)	C, Q and Validation (V)	Analysis of new traces and classification	No
GPS5	UML	No visualization	C	Not provided	Yes
GPS6	XMI (Ecore)	Graphical models (EMF, AMW)	C	Not provided	Yes
GPS7	UML, relational schemas (Rational Rose), XML Schema and EMF	Graphical models	C, U, D and Q	Not provided	Yes
GPS8	UML and EMF	GEF3D models	C and Q	Not provided	Yes
GPS9	EMF	Graphical models (ModeLink)	C, U, D and Q	Not provided	Yes
GPS10	XML	HTML report	C and U	Traceability report	Yes
GPS11	EMF, UML	Graphical representation and HTML report	C and Q	Traceability report	Yes

GPS	Model languages supported	Information is visualized by…	Operations supported	Analysis of traces	Has been implemented?
GPS12	EMF	Not provided	None	Not provided	Yes
GPS13	XMI (Ecore)	Graphical representation (TAP, UML and GMF)	C, U, D and Q	TAP provides model coverage and orphan and impact analyses	Yes
GPS14	UML	UML model	C, U and Q	Not provided	No
GPS15	Not defined	Not defined	C, U, D and Q	Some analysis can be implemented based on queries	No
GPS16	Kermeta and XMI models	Graphviz representation	C	Not provided	Yes
GPS17	Not defined	XML Model	C	Not provided	No

These data allow us to state that the main topics covered by the proposals contained in the primary studies are: traces generation (automatic and/or manual), metamodel (general or specific purpose), traces management (storage, visualization, operations supported and traces analysis) and implementation (complete toolkit or partial). Figure 5 illustrates this statement observation by means of a feature diagram [56].

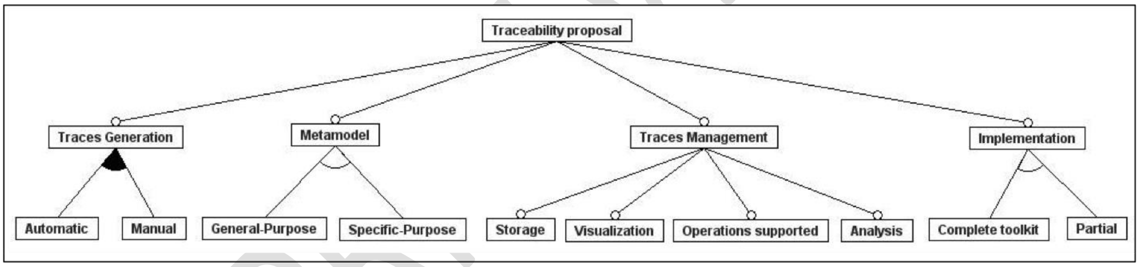

Figure 5. Feature diagram representing the main issues covered by traceability proposals

In particular, each of the topics related to traces management (storage, visualization, operations supported and traces analysis) comprise another set of topics, as shown in Figure 6. For instance, each proposal can support a different set of operations (creation, deletion, modification and/or retrieval).

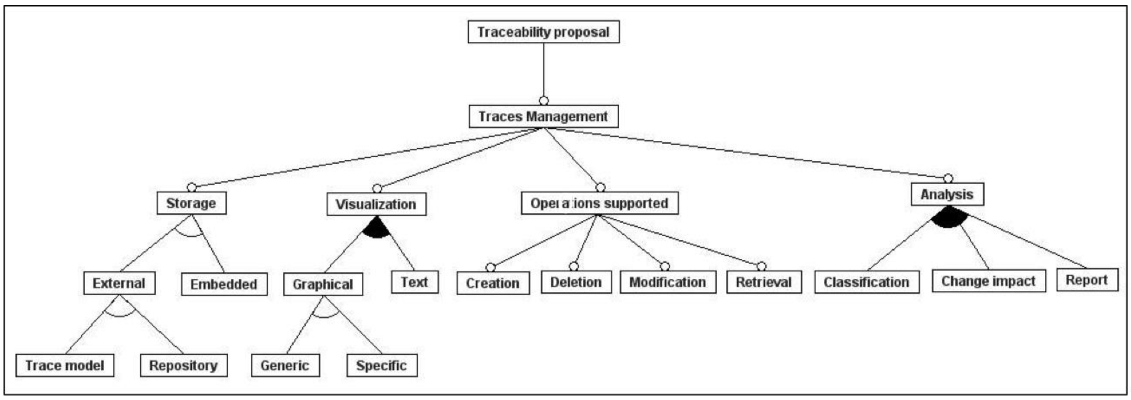

Figure 6. Features of traces management

3.3.1.GPSs assessment results

The next step was to assess the quality of each GPS by assigning a value to each of the GPS assessment questions posed in Section 2.5.1. The score assigned to each study for each question is shown in Table 8.

The results show that all of the GPS got some points, while only two of the GPS scored less than 2.5 points, two reached 6 or more points and four out the seventeen GPS are located in the statistical mode (that is 4). The last column ("% max GPS") shows the percentage reached by each GPS over the total score (i.e., 7). The penultimate row ("% total score") shows the percentage of points obtained by all the GPS with regard to the total number of points obtained by all the GPSs in all the Quality Assessment questions. Finally, the last row ("% max GA") corresponds to the percentage of points collected by the values assigned for a given Quality Assessment question over the points that would be collected if every GPS got the highest score (i.e. 17 * 1 = 17). The arithmetic mean of the scores is 3.97 and the standard deviation 1.32.

Table 8. Quality evaluation of GPSs.

ID	GA1	GA2	GA3	GA4	GA5	GA6	GA7	Total Score	% max GPS
GPS1	P	Y	P	Y	Y	N	P	**4.5**	**64.29%**
GPS2	P	Y	Y	Y	Y	Y	Y	**6.5**	**92.86%**
GPS3	P	Y	Y	P	N	N	Y	**4**	**57.14%**
GPS4	N	Y	Y	P	Y	P	N	**4**	**57.14%**
GPS5	P	Y	N	P	N	N	Y	**3**	**42.86%**
GPS6	Y	Y	P	P	N	N	Y	**4**	**57.14%**
GPS7	Y	Y	Y	Y	N	N	Y	**5**	**71.43%**
GPS8	P	Y	Y	P	N	N	Y	**4**	**57.14%**
GPS9	Y	Y	Y	Y	N	N	P	**4.5**	**64.29%**
GPS10	P	Y	Y	P	N	Y	P	**4.5**	**64.29%**
GPS11	P	Y	Y	P	N	Y	Y	**5**	**71.43%**
GPS12	N	N	N	N	N	N	Y	**1**	**14.29%**
GPS13	Y	Y	Y	Y	P	Y	P	**6**	**85.71%**
GPS14	N	Y	P	P	N	N	N	**2**	**28.57%**
GPS15	N	Y	N	Y	N	P	N	**2.5**	**35.71%**
GPS16	P	Y	Y	P	N	N	P	**3.5**	**50.00%**
GPS17	Y	Y	P	P	N	P	N	**3.5**	**50.00%**
Total	**9**	**16**	**12**	**11**	**3,5**	**5,5**	**10,5**	**67,5**	
% total score	**13.33%**	**23.70%**	**17.78%**	**16.30%**	**5.19%**	**8.15%**	**15.56%**	**100 %**	
% max GA	**52.94%**	**94.12%**	**70.59%**	**64.71%**	**20.59%**	**32.35%**	**61.76%**		

The GPS that obtained the highest score is GPS2 with a score of 6.5, which represents about 95% (92,86%) of the maximum possible. By contrast, the GPS12 obtained a score of 1, representing less than 15% (14.29%) of the maximum score that one GPS could get. Figure 7 shows a pie chart depicting the distribution of scores for the GPS assessment questions. It illustrates that the first four questions are distributed over 70% (71.11%) of the total score, while questions 5 (exchange) and 6 (analysis) represent less than 15% (13.33%).

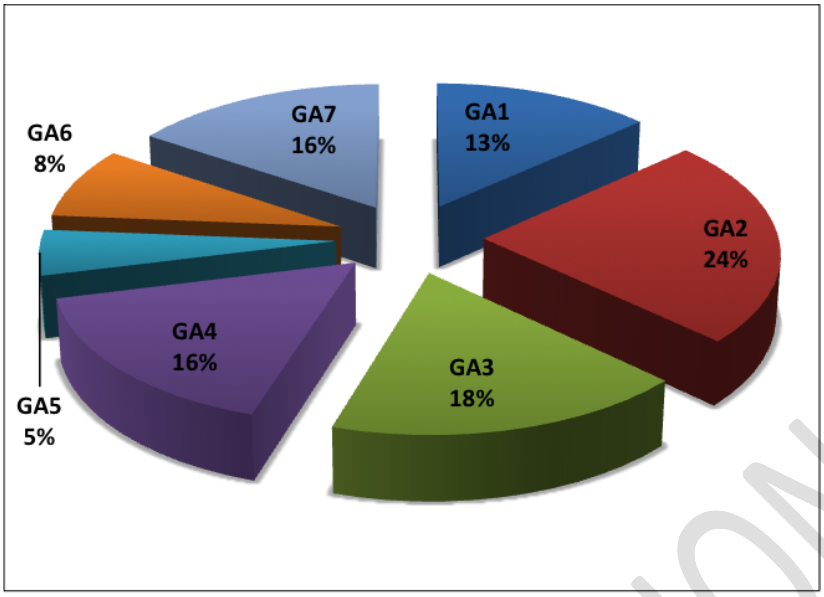

Figure 7. Scores for GPS assessment questions

The proposal that obtains the highest score is GPS2, since it provides answers for all the aspects evaluated in this work. From our point of view, this is the most mature work. In contrast, the proposal that obtains the lowest score is GPS12. These works focus primarily on generating labels with which to maintain the traceability between models and code in the context of M2T transformations in the Eclipse Modeling Framework (EMF). They do not therefore consider the visualization or storage of trace links.

Overall we can state that the results highlight that the aspects which are most commonly addressed by the works reviewed are those that provide the basis with which to address advanced tasks that have yet to be addressed. In other words, basic operations such as storage, creation, modification, removal of trace links are widely covered. In contrast, the aspects that are in some way derived from these basic aspects are less commonly addressed. For instance, it is not possible to analyse traceability information if trace links are not previously stored, or retrieval operations are not supported.

3.4. Classification factors

In this review we were also interested in finding out if there was any relationship between the quality of the reviewed studies and the type of forum where they were published.

To begin with, Table 9 simply collects the type of publication in which each Primary Study was published. The first column shows the acronym for the publication, whose name is shown in the second column and whose type is shown in the third column: (C) conference, (B) book, (J) journal or (TR) technical report. In order to assist the assessment, a fourth column (a so called "Level") shows how the publication is ranked according to the reference rankings for research in computer science: the CORE ranking for conferences and the JCR ranking for journals. For the remaining types (book and technical report), the value is set at "Non Applicable" (N/A). Finally, the last column ("#PS") shows the number of primary studies published in each of the publications listed.

Table 9. Number of primary studies for the different publications

Acronym	Publication	Type	Level	# PS
SAC	ACM Symposium on Applied Computing	C	CORE-B	2
ASDLS	Applications of Specification and Design Languages for SoCs	B	N/A	1
ECMFA	European Conference on Modelling Foundations and Applications [3]	C	NO-CORE	15
IEEE-TC	IEEE Transactions on Computers	J	JCR-Q2	1
INFSOF	Information and Software Technology	J	JCR-Q1	1
ICSEM	International Conference on Systems Engineering and Modelling	C	CORE-C	1
MT-ATL	International Workshop on Model Transformation with ATL	C	CORE-B	1
MODELS	Model Driven Engineering Languages and Systems	C	CORE-B	1
PRIMIUM	Process Innovation for Enterprise Software	B	N/A	1
SOSYM	Software and Systems Modelling	J	JCR-Q2	2
SLE	Software Language Engineering	C	CORE-B	1
TR-GRACE	Technical Report GRACE	TR	N/A	1
UO	University of Olso, Norway	TR	N/A	1

In order to determine if there is a relationship between quality scores (QAs and GAs) and the type and ranking of the publications, we needed to distinguish the number of each type of publications included in each GPS. Therefore, the grey-shadowed columns of Table 10 show how many Primary Studies of the different types are included in each GPS. Additionally, it shows the ranking (JCR and CORE) of the Primary Studies that compose each GPS. Note that we found no primary studies that fell in the CORE-A*, CORE-A, JCR-Q3 and JCR-Q4 scores. Finally, in order to better relate quality and GPS assessment with the type of publication, the two last columns retrieve the score of each GPS for the quality questions (QAs and GAs) that were previously presented in Table 5 and Table 8 respectively.

[3] Note that, prior to 2010, ECMFA was called ECMDA-FA (European Conference on Model Driven Architecture: Foundations and Applications

Table 10. Primary Studies by type and ranking of the publication

ID	# PS	J	JCR-Q1	JCR-Q2	C	CORE-B	CORE-C	NO-CORE	B	TR	QA Score	GA Score
GPS1	1	0	0	0	1	0	0	1	0	0	4	4.5
GPS2	2	1	0	1	1	0	0	1	0	0	4.5	6.5
GPS3	1	0	0	0	0	0	0	0	1	0	4.5	4
GPS4	5	1	0	1	4	1	0	3	0	0	5	4
GPS5	1	0	0	0	1	0	0	1	0	0	4	3
GPS6	6	0	0	0	6	3	1	2	0	0	4.5	4
GPS7	1	0	0	0	1	0	0	1	0	0	3.5	5
GPS8	1	0	0	0	1	0	0	1	0	0	4.5	4
GPS9	1	0	0	0	1	1	0	0	0	0	4.5	4.5
GPS10	1	1	1	0	0	0	0	0	0	0	4	4.5
GPS11	1	1	0	1	0	0	0	0	0	0	4.5	5
GPS12	1	0	0	0	0	0	0	0	0	1	3	1
GPS13	3	0	0	0	2	0	0	2	0	1	4.5	6
GPS14	1	0	0	0	1	0	0	1	0	0	4	2
GPS15	1	0	0	0	1	0	0	1	0	0	3	2.5
GPS16	1	0	0	0	1	0	0	1	0	0	3.5	3.5
GPS17	1	0	0	0	0	0	0	0	1	0	3	3.5
TOTAL	29	4	1	3	21	5	1	15	2	2	68.5	67.5

Furthermore, Table 11 shows the amount of GPSs that included at least one publication in a journal, conference, book and/or technical report and the average of the scores (from Quality and GPS Assessment) achieved by the GPSs fulfilling these criteria.

Table 11. Average scores for QAs and GPS questions by type of publication

	GPS including at least one publication in …	Average score from QAs	Average score from GAs
Journal	4	4.50	5.00
Conference	12	4.13	4.13
Book	2	3.75	3.75
Technical Report	2	3.75	3.50

Note that the category "journal" achieves the best quality score average (4.50 for the QAs and 5.00 for the GAs) and the "technical report" category attains the worst (3.75 for the QAs and 3.50 for the GAs). The results obtained correspond to what might be expected according to the review process of each type of publication, i.e. in most cases, journals and conference review processes are more exhaustive and in-depth than those of books and technical reports which, in some cases, are only reviewed by their own authors.

The next issue was to study the relationship between the year of publication and the type and/or level of publication. Table 12 shows the publications per year, divided by the publication categories. These data allow us to note that 2006 and 2010 were the most productive years (6 and 7 primary studies, respectively). With regard to conferences, 2009 and 2010 were the years in which there were most publications in mid-level conferences (CORE-B). With regard to journals, the most productive year was 2011 (we should remind the reader that we have retrieved studies published before March 2011) and we can verify that publications in high-impact journals occurred in 2009 and 2011. These points serve to illustrate that the number of relevant publications (journals and CORE-B publications) on the topic is an increasing trend. In general, high-level publications have been published more recently, which allows us to conclude that the area has probably attracted more interest during the course of the past few years and the results are becoming mature enough to be published in reference forums.

Table 12. Number of publications by year, type and level of the publication

Type-Level /Year	**2005**	**2006**	**2007**	**2008**	**2009**	**2010**	**2011**	**Total**
Book		**1**			**1**			**2**
Conference	**3**	**5**	**3**	**3**	**2**	**5**		**21**
CORE – B		1			2	2		**5**
CORE – C			1					**1**
NO-CORE	3	4	2	3		3		**15**
Journal					**1**	**1**	**2**	**4**
JCR-Q1					1			**1**
JCR-Q2						1	2	**3**
Tec. Report			**1**			**1**		**2**
Total	**3**	**6**	**4**	**3**	**4**	**7**	**2**	**29**

We believe that, although traceability management in MDE has gained attention since the beginning of MDE, it was not possible for works in the area to become mature until MDE-based tools had become sufficiently mature. As MDE proposals have gained a certain level of maturity in the last few years, works on traceability management have therefore benefitted from it and have also gained a certain level of maturity. Their quality has consequently increased, and they are the subject of relevant publications in the area.

4. Discussion

In this section, we will answer the research questions posed in Section 2.1, using the main results just presented in section 3.

4.1. What level of automation is suggested by methodological proposals for the generation of trace links?

As we have mentioned before, automation is one of the keys to MDE [57]. Consequently, there is a need to take advantage of the new landscape that MDE offers for the management of traceability. In particular, the automatic generation of trace links between different software artifacts becomes more feasible since most of those assets are models produced by means of model transformations.

Following on from this idea, it seems to be widely acknowledged that the best way to automate the generation of trace links at model level consists of being able to derive them from relationships specified at metamodel level. Ideally, we should be able to identify these metamodel-level relationships by means of any automatic technique as well.

In order to evaluate the proposals according to this research question, we have analyzed the results of the Quality Assessment question 1 (GA1) that checks whether the proposal suggests any techniques to improve the level of automation for the generation of trace links. We have found that four Groups of Primary Studies, namely GPS4, GPS12, GPS14 and GPS15, which constitute 23.53% of the total number of GPS, do not consider the automation of trace links generation at all, i.e. they consider it to be a task to be completely performed manually.

The rest of GPS (76.47%) either put forward or consider a technique to improve the automation of trace links generation. More concretely, eight GPS (47.05% of the total), those that received a "Partly" score for GA1, propose the generation of trace links from relationships specified at metamodel level, while five GPS (29.41% of all GPS) also suggested techniques to automate the identification of this metamodel-level relationships. This is the expected scenario, since a priori generating model-level trace links is much simpler than identifying metamodel-level relationships.

Following on from this, we will now go into the suggestions for automatic generation of trace links at model level found in the proposals analyzed. Since model transformations, being the main technological bridge between models [21, 58], are considered the key to increasing the level of automation of any model-based proposal [22], eleven of the analyzed proposals (64.70%) lean on model transformations to generate trace links. However, these transformations are dependent on a specific transformation engine. As a result, most of the suggestions for trace links generation cannot be adapted to different model transformation engines. For instance, if a given proposal is focused on generating trace links from graph-based transformations, adapting its generation technique to declarative transformations might result in a too complex and tedious task (when feasible). A mechanism for trace links generation that does not depend on any specific transformation engine would alleviate this problem. However, the proposals reviewed are engine-specific, and we thus consider that there is a gap in terms of technology-agnostic mechanisms for trace links generation.

Regarding the definition of traceability relationships at metamodel level, we have found four proposals (23.52%) which require the user to do it manually, four proposals (23.52%) which provide techniques to generate them automatically and five proposals (29.41%) which put forward ideas to combine both approaches.

Of the proposals that follow a manual approach, three of them (GPS5, GPS10 and GPS16) suggest that the user has to add traceability information manually to existing model transformations, so that not only the target model but also traceability information is generated by this transformation (whether in a trace model or in the target model itself); GPS11 in turn suggests the definition of an initial traceability model from which subsequent traceability models are generated. Since the user has to provide it with some information to start the process, we argue that these proposals do not consider techniques that completely automate the generation of trace links.

By contrast, GPS7 and GPS9 suggest that the metamodel-level relationships, from which trace links will be later generated, could be derived from previously defined metamodel-level relationships. The purpose of defining the latter did not only have to specify traceability relationships but perform any model-processing task, such as model comparison or model validation. On the other hand, the suggestion of GPS8 and GPS17 is to obtain the model-level trace links from the implicit trace relationships handled by any model transformation engine when a model transformation is executed. Again, this results in particular solutions for each specific transformation engine. Adapting the mechanism for trace links generation to any other transformation engine might be a complex task.

Finally, a set of proposals (GPS1, GPS2, GPS3, GPS6 and GPS13) combine the aforementioned ideas for the identification of metamodel-level relationships. The ideas collected in these proposals cover a wide scope. Some of them (GPS1 and GPS13) suggest that the best way to generate trace links is to use the implicit metamodel-level relationships collected in any model transformation. Thus, they suggest extending the transformation engine used in order to make explicit the internal traceability model used by the engine during the execution of the transformation. Thus, automatic generation of trace links depends on whether the selected engine provides an internal traceability mechanism. On the other hand, GPS2 provides the user with a generic framework for traceability management that imports existing traceability artifacts. So, how traceability relationships are identified as well as how trace links are generated depends basically on the framework from which such artifacts were imported. Additionally, it allows the user to specify such relationships manually. GPS3 proposes that the user explicitly defines some relationships, while some other implicit relationships, derived from previous ones, will be created automatically (GPS3). Finally, GP6 argues in favor of combining the provision of some mechanism to extract explicit relationships from the information contained in model transformation rules, with some mechanism to define trace links by hand.

To sum up, according to the data collected in Table 8, we have found that thirteen out of the seventeen GPS analyzed (76.47%) suggest some automatic or semi-automatic technique for the generation of traceability information. Therefore, we might conclude that automatic generation of traceability information is a widely recognized issue that should be considered by any proposal for traceability management in MDE. Nevertheless, most of the proposals are focused on the generation of trace links at model level whereas the automatic identification of relationships at metamodel-level is poorly covered.

4.2. How do methodological proposals suggest that traceability should be managed and analyzed?

As a way to answer the research question posed, this section revisits the results obtained for the GPSs Assessment questions referred to in Section 2.5.1. The questions were focused on studying the different Groups of Primary Studies according to their proposals for storage (GA2), visualization (GA3), of CRUD operations (GA4) and analysis (GA6) of traceability information, we will answer the question raised with the help of the obtained results.

With regard to storage, in Section 3.3.1 we stated that all the GPS (except GPS12) provide some kind of storage mechanism. Indeed, the accumulative score for this question reached 94.12% of the maximum value that a Quality Assessment question could get. Table 6 provides you with detailed information on this matter by enumerating which is the storage mechanism used by each GPS. From analyzing this data, we found that the preferred form of persisting traceability information is by means of trace models.

Likewise, most of the GPS analyzed included some contribution regarding visualization of traceability information. In fact, question 3 reached 70.59% of the maximum value for a Quality Assessment. In particular, there were just three proposals that did not consider the visualization issue at all, while four proposals made a partial contribution to this task (GPS1, GPS6, GPS14 and GPS17) by using XML, EMF, AMW and UML models for visualization. Other proposals, such as TraVis (GPS3) ModelLink (GPS4) and TAP (GPS13) opted for graphical models, whereas GPS2, GPS10, and GPS11 opted for textual representations.

Concerning CRUD operations, Table 7 collects the data about which operations are supported by each GPS. We found six proposals to support the four operations (create, retrieve –query, update, and delete), ten proposals that support at least one operation and only one proposal (GPS12) that does not support any CRUD operation. The quality score obtained for this Quality Assessment question (GA5) represents 64.71% of the maximum value that one GA could get.

Finally, only four proposals deal with the analysis of the traceability information obtained. With regard to the rest of the proposals, though three of them do consider the importance of performing some kind of analysis with the traceability information gathered, none of them provide pointers in this sense, while the other ten proposals do not consider this issue at all. The percentage obtained by Quality Assessment question GA6 (32.35%) clearly suggests that this is a future line for research, which is still quite immature. As future research focused on this issue, we could point researchers in the direction of the ideas presented in the few proposals that actually deal with the topic. In those works, trace links were analyzed with the following purposes: impact analysis (GPS2 and GPS13), traceability reports (GPS10 y GPS11) and coverage analysis plus orphans analysis (GPS13).

4.3. Are there tools or frameworks that provide technological support for the management of traceability in the context of MDE?

As has been mentioned on several occasions throughout this paper, from a theoretical point of view MDE provides a new landscape in which to improve the management of traceability in software development. Despite the traditional gap between theoretical ideas and practical deployment that appears by many Software Engineering proposals, the scores for the Quality Assessment question 7 presented in Table 7 allow us to argue that this is not the case of traceability management in MDE. Only four Groups of Primary Studies (namely GPS4, GPS14, GPS15 and GPS17) provide the user with no suggestions for the implementation of their proposals.

By contrast, five Groups provide a partial implementation (GPS1, GPS9, GPS10, GPS13 and GPS16) while eight Groups provide a complete implementation (GPS2, GPS3, GPS5, GPS6, GPS7, GPS8, GPS11 and GPS12). It is worth noting that most of these implementations have been developed using Eclipse IDE and more specifically the technologies provided in the context of the Eclipse Modelling Project (EMP)[4] that collect a set of frameworks, tools and reference implementations for standards that help with the development of technological support for any methodological proposal based on MDE principles.

Therefore, the main conclusions for this research question are that it is feasible to develop technical support for traceability management tasks in MDE proposals and that existing works in this sense have commonly adopted EMF-based technologies as standard de-facto for this purpose.

4.4. What are the limitations of state of the art in traceability management in the context of MDE?

Analyzing the discussion about the aforementioned research questions (RQ1, RQ2 and RQ3), we have identified some limitations and points for possible improvements of state of the art traceability management in MDE.

In section 4.1, we pointed out that many proposals lean on model transformations to generate trace links automatically (or semi-automatically). In that section, we mentioned that these transformations are defined atop a specific model transformation engine. Consequently, they are dependent on this transformation engine. However, we have to take in to account that model transformations play a key role in Model-Driven Software Development. As a result, many languages and tools that develop model transformations have come to the fore during the last number of years [59]. Therefore, due to the fact that there are many alternatives to develop model transformations and it is an emerging and evolving field, it may be desirable to bundle

[4] Eclipse Modeling Project website: http://eclipse.org/modeling/

the transformations in a given proposal that does not depend on a specific model transformation language or engine. This would allow us to adapt the proposal to advances made in the field, such as improved or evolved model transformation languages. In other words, swapping the transformation technology or approach would not imply losing the possibility of generating trace links.

Another issue that leaves room for improvement is the storage and processing of traceability information. Regarding the former we have found that though some Groups of Primary Studies argue in favor of using trace repositories, most of them use models to store trace links. The use of models as trace containers allows for the application of MDE principles to process trace links, just as we do with any other model. That is to say, we can transform, validate, serialize, merge or compare them using any of the existing tools that provide us with technological support for these tasks. On the other hand, using models to manage traceability information implies the need for a traceability metamodel. In this regard, we have found that most of the reviewed proposals opted for defining their own metamodel, which leads to a huge problem of interoperability: trace links cannot be interchanged among different proposals. As Table 8 showed, Quality Assessment question 5 focused on finding contributions to the exchange of traceability information, obtained the lowest score amongst all the Quality Assessment questions posed.

Somehow related with the use of models that store trace links is the fact that some proposals do not provide the user with a specific tool or technique to edit or display traceability information. They use the editors and visualization mechanism bundles in the underlying (meta-) modeling frameworks, such as EMF or UML toolkits. Unfortunately, the generic nature of such editors is not the best way to cope with the specific nature of trace models. For instance, some of them only support the representation of trace objects, without the ability of navigating them to reach the *traced* objects while some others need to load the whole model to support the navigation to those *traced* objects. Indeed, some of the Groups of Primary Studies reviewed (GPS3, GPS4 and GPS8) bring this problem to our attention and provide us with specific tools and techniques to edit and display traceability information.

Another point for improvement is related to CRUD operations. Despite having mentioned on numerous occasions that the only way in which to obtain a full return of MDE promises is by leveraging the level of automation, we should consider that complete automation is not always feasible or even recommendable. Automated tasks do not always deliver the expected result, and manual refinement mechanisms should therefore also be supported. In this respect, we have detected that some proposals do not consider the complete set of CRUD operations for traceability information, which prevents the manual refinement of the trace links that implement the traceability information available. Additionally, the absence of proper querying mechanisms hampers the analysis of traceability information. The absence of support for a complete set of

CRUD operations and proper querying mechanisms has therefore also been identified as directions for future work.

Regarding the analysis of traceability information, in spite of the fact that some evidence exists about the importance of analyzing traceability information to carry out other software activities [4, 5, 12, 14, 16], among the Groups of Primary Studies reviewed, only four of them provide the user with contributions in this regard. In fact Quality Assessment question 6, which checked if the different proposals perform some kind of analysis with traceability information, got the second lowest score (5.5 out of 17). This leads us to conclude that this issue has not been properly addressed in state of the art traceability management in the context of MDE.

4.5. Are there forums (e.g. journals or conferences) that specialize in dealing with traceability management in MDE?

To help address this question, Figure 8 shows three pie charts. The first one (a) shows the distribution of primary studies across the different publications. We observed that 51.72% of the Primary Studies were published in the ECMFA conference. Apart from this, the distribution of publication forums for the rest of Primary Studies is very disperse, with only two forums (SAC and SOSYM) accounting for more than one publication.

Figure 8(b) shows the distribution of Primary Studies with regard to publication types. Note that the preferred places in which to publish contributions on traceability management in MDE have been conferences (72%). It seems logical to submit proposals on recent topics in order to obtain feedback on them. We therefore believe that this is still an emerging area.

Finally, Figure 8(c) sums up the data related to the relevance of forums, as discussed in Section 3.4. More than half of the publications were published in conferences that were not ranked in CORE, whereas some recent publications have been published in CORE-B conferences (17%), and JCR-indexed journals from the first and second quartiles (11.4%).

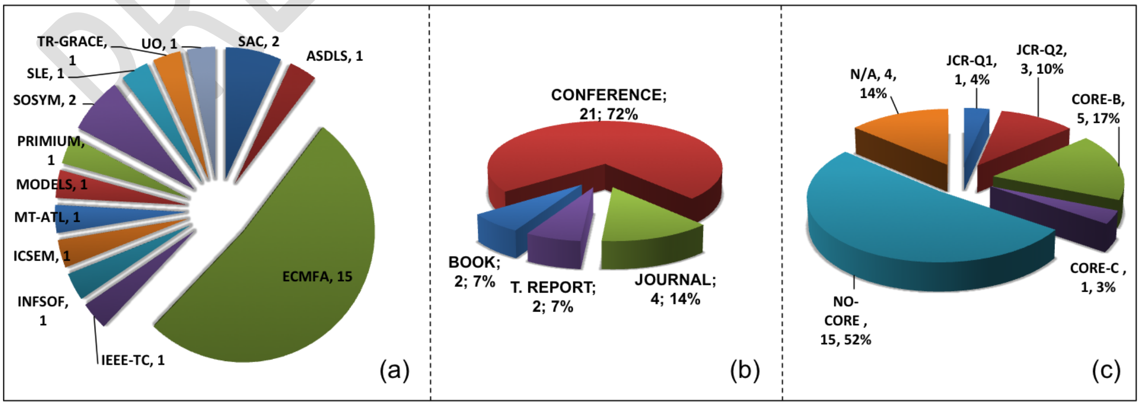

Figure 8. Distribution of publications: (a) by publication; (b) by type of publication; (c) by ranking of publication

Overall, the high number of results published in conferences and the low number of publications in high-impact journals might indicate that many proposals are still under development and that their authors are seeking rapid feedback with which to improve them. On the other hand, as mentioned in Section 3.4, the best-ranked publications have mainly appeared in the last few years. This might suggest that the maturity level reached by MDE technology [60, 61] has enabled researchers to address more ambitious proposals, such as traceability management.

4.6. Discussion about the limitations of this Systematic Literature Review.

As mentioned previously, we have followed the advice and guidelines proposed by Kitchenham [28] and Biolchini et al. [26] to conduct this systematic literature review. However, we have deviated slightly from those guidelines in the following points of the process:

- We did not execute manual searches.

  Section 2.2 described the search process followed in this study, which consists mainly of a set of searches executed in several digital libraries. Therefore, no manual search has been carried out. This deviation might miss some studies published in journals or conferences which are not available on-line.
- We did not use the PICOC (Population, Intervention, Comparison, Outcome, Context) criteria to define the research questions.

  In spite of the fact that we did not use these criteria to define the research questions, they can answer some of these criteria. For example, the context of our research questions is 'MDE' and the population is 'traceability management'. Although this deviation may implicate that research questions can be somehow unstructured and subjective, we would like to stress that the research questions posed are the result of an iterative process where they have been continuously improved and refined.
- This study has not been reviewed by external experts.

  Despite the fact that we did not contact external experts to evaluate this review as-is, initial versions of some of the main results and conclusions of this work were presented and discussed in the most important Spanish conference about Software Engineering and Databases (JISBD 2011) [62, 63]. Moreover, during the development of this study, some authors have played the role of researchers whereas some others acted as referees, therefore providing some kind of internal evaluations. Nevertheless, we cannot discard the possibility of there being some mistakes in the information extracted or the study containing researcher bias.
- We did not record the divergences between reviewers.

  In several cases, reviewers did not agree on the information extracted from the studies and those divergences were not formally recorded, though the versions repository that has been used during the development of this study allows us to recover any previous observation or conclusion and reformulate the study accordingly. Furthermore, all of the divergences were solved after a discussion process that, apart from enriching the study, led to a consensus on the given issue.

5. Related Works

In this section, we will briefly present some related works [9, 64-66]. In particular, those works that focused on performing some type of review of the literature on the topic covered in this study. Hence, the works considered in this section were selected as relevant studies during the primary studies selection phase of the systematic literature review process (see Figure 3(b)), but were excluded by the exclusion criteria, since they are studies whose main purpose is to classify other articles or they are systematic literature reviews themselves (we considered these articles as secondary studies, see Section 2.3).

One of the most referenced works in the context of traceability in MDE is the one from Aizenbud-Reshef et al. [9]. In some sense, this paper introduces the concept of model traceability. The authors review the current state of the art of traceability and its possible synergy with MDE, highlighting the main problems. In particular, they state that one of the main problems for the adoption and effective use of traceability is the overheads incurred in manually creating and maintaining relationships. Consequently, the paper reviews the latest advances on the automatic discovery of trace relationships, highlighting the fact that MDD provides new opportunities for establishing and using traceability information. In this line, they discuss the automatic generation of trace information through transformations and the use of traceability relationships for consistency maintenance and model synchronization. Regarding the main differences relevant to the study presented here, it could be said that the work from Aizenbud-Reshef et al. is located at a higher abstraction level. While they focus on identifying and presenting the generic advantages that MDE can bring to the management of traceability, our work focused on assessing to which point these advantages have been put into practice and, if applicable, how they have been carried out. For instance, by looking at which are the preferred ways of generating trace links or the most adopted approaches for storing such trace links. Furthermore, since Aizenbud-Reshef et al's study was conducted in 2006, it does not include more recent works in the field. Indeed, 20 out of the 29 primary studies that were used in our research were published after 2006.

In [64] Galvao and Goknil present a study where five criteria are evaluated over different MDE proposals. These criteria are: representation of traceability information, mapping, scalability, change impact analysis and supporting tools. Although they do provide us with a complete study on the topics covered, they do not consider management operations, such as creation, deletion, querying and/or update of trace links. Neither storage, analysis of traceability information, nor the generation of trace links is evaluated.

In turn, Schwarz et al. present in [65] a work an overview of how traceability is considered in the software development process. The paper is only organized according to six operations: definition, recording (storage), identification, maintenance, recovery (querying) and/or use of traceability information in specific scenarios. Therefore, despite being a comprehensive study, it does not deal with two very relevant issues which are very important in the research objective of this study: tool support and the analysis of traceability information.

Finally, another relevant contribution is that by Winkler and Von Pilgrim [66] that offers an overview of current research and practice of traceability and requirements engineering, suggesting that MDD could help to address some problems in both of these areas. As part of the study, concordances and differences between these areas are pointed out, as well as new challenges being identified. The work is structured according to two main directions. On the one hand, it describes the possibilities provided by MDD for the automatic recording of traceability information. On the other hand, it offers a comprehensive view of how traceability is handled in requirements engineering. This work provides a valuable "body of knowledge" by analyzing the major conferences, workshops and journals on requirements engineering and MDD. The information presented is classified into four categories: basic, working with traces, practice and solutions. With regard to working with traces in MDE, this paper focuses on the generic use of traceability information, its visualization and its usability (an interesting topic not really covered by our study). However, two main issues are not covered by this work: it does not address the exchange and analysis of traceability information. These issues have been identified in this study as the most immature operations as regards traceability management in the MDE context.

In conclusion, two big differences have been found between these works and the study presented here. On the one hand, these works do not follow a rigorous process of review of the literature, like a systematic literature review. On the other hand, the research criteria are different from that evaluated here.

6. Conclusions

Traceability information has been traditionally acknowledged to play a cornerstone role for the development of different Software Engineering activities, such as impact analysis, regression testing or maintenance in general. Unfortunately, the lack of automatic or semi-automatic support has hampered the issue of maintaining links among software artifacts, which becomes a tedious and time consuming task [17-20]. The key role of models in MDE proposals throughout the development process can facilitate the recording and maintenance of traceability information. Hence, given the relevance of traceability in software development and the appearance of a new scenario that could improve the research and practice of traceability management, we have conducted a systematic literature review of the literature on the topic based on the guidelines proposed by Kitchenham [28] and Biolchini et al. [26].

The figures of this review can be summarized as follows: the execution of searches (using adapted query strings) in each search engine considered raised 10028 results, from which just 267 were identified as relevant studies. After eliminating duplicates, the 157 non-duplicated studies were evaluated according to the exclusion criteria defined. As a result, only 29 works became primary studies. Finally, since some of them were evolutions or variations of the same idea, the 29 primary studies were grouped into 17 Groups of Primary Studies which were used to conduct the rest of the review process, whose main goal was to identify and analyze the state of the art in traceability management in the context of MDE.

The results of this review have permitted us to gather certain data and ideas which have been presented in the discussion section in order to provide a complete overview of the state-of-the-art in the area. This review has additionally served to identify some of the main challenges that the management of traceability in MDE should tackle in the coming years. Some of these might be:

- With regard to quality assessment, the specification of the research method followed in the works evaluated is a major handicap. None of the proposals reviewed explicitly specified the research method followed. The other quality issues evaluated have acceptable quality ratings such as the definition of the research goal or comparison to other related works.
- Some proposals which use model transformations to generate trace links have been identified. However, they are technology-dependent since they were built according to the features of a specific transformation engine. It would therefore be appropriate to provide a technology-agnostic means of generating trace links from model transformations in order to permit their application to any existing model transformation language.
- The fact that many proposals have opted to use their own traceability metamodel which results in a problem of information interchange.
- Despite most of the reviewed proposals providing some kind of support for visualizing traceability information, they are mainly ad-hoc mechanisms that do not fit the specific needs of visualizing traceability information.
- Even though supporting CRUD operations for handling traceability information is partially supported by the majority of the proposals, very few of them support the complete set of CRUD operations. This raises two main issues: firstly, the user is not always capable of refining the traceability information generated (semi-)automatically; secondly, the lack of querying support hampers or inabilities the analysis of traceability information.

To conclude, we would like to stress an idea that has been present throughout this study: we are now in a position to face the development (or refinement) of methodological and technical proposals for dealing with traceability in MDE. Since MDE technology is becoming more mature and stable, it is time to face the different challenges that will allow us to make the most of traceability information. Maybe we can develop some of the sound principles related with using traceability information that, according to traditional Software Engineering, should drive any software development proposal.

Acknowledgements

This research was carried out in the framework of the MASAI project (TIN-2011-22617) and the Technical Support Staff Subprogram (MICCINN-PTA-2009), which are partially financed by the Spanish Ministry of Science and Innovation.

References

[1] IEEE Standard Glossary of Software Engineering Terminology. IEEE Std 610.12-1990, 1990: p. 1.

[2] B. Ramesh, C. Stubbs, T. Powers, M. Edwards, Requirements traceability: Theory and practice. Annals of Software Engineering, 1997. 3: p. 397-415.

[3] H. Asuncion. Towards practical software traceability. In The 30th International Conference on Software Engineering. 2008. Leipzig, Germany: ACM.

[4] H. Schwarz, Taxonomy and definition of the explicit traceability information suppliable for guiding model-driven, ontology-supported development. Project Deliverable ICT216691/UoKL, 2009, WP4-D1/D/PU/b1, MOST Project.

[5] P. Valderas, V. Pelechano, Introducing requirements traceability support in model-driven development of web applications. Information and Software Technology, 2009. 51(4): p. 749-768.

[6] S. Walderhaug, E. Stav, U. Johansen, G.K. Olsen, Traceability in Model-Driven Software Development. In Designing Software Intensive Systems: Methods and Principles, P. Tiako, Editor, 2009. p. 133-159.

[7] L.C. Briand, Y. Labiche, L. O'Sullivan. Impact analysis and change management of UML models. In Proceedings of International Conference on Software Maintenance, ICSM. 2003.

[8] A.v. Knethen, M. Grund. QuaTrace: A Tool Environment for (Semi-) Automatic Impact Analysis Based on Traces. In Proceedings of the International Conference on Software Maintenance. 2003. IEEE Computer Society.

[9] N. Aizenbud-Reshef, B.T. Nolan, J. Rubin, Y. Shaham-Gafni, Model traceability. IBM Syst. J., 2006. 45(3): p. 515-526.

[10] P. Campos, N.J. Nunes, Practitioner Tools and Workstyles for User-Interface Design. IEEE Softw., 2007. 24(1): p. 73-80.

[11] R. Paige, G.K. Olsen, D. Kolovos, S. Zschaler, C. Power. Building model-driven engineering traceability classifications. In Proceedings of the 4th European Conference on Model Driven Architecture - Traceability Workshop (ECMDA-TW'08). 2008. Berlin, Germany.

[12] L. Naslavsky, T.A. Alspaugh, D.J. Richardson, H. Ziv. Using scenarios to support traceability. In Proceedings of the 3rd international workshop on Traceability in emerging forms of software engineering. 2005. Long Beach, California: ACM.

[13] R. Brcina, M. Riebisch. Defining a traceability link semantics for design decision support. In Proceedings of the 4th European Conference on Model Driven Architecture - Traceability Workshop (ECMDA-TW'08). 2008. Berlin, Germany.

[14] N. Anquetil, U. Kulesza, R. Mitschke, A. Moreira, J.-C. Royer, A. Rummler, A. Sousa, A model-driven traceability framework for software product lines. Software and Systems Modeling, 2010. 9(4): p. 427-451.

[15] A. Sousa, U. Kulesza, A. Rummler, N. Anquetil, R. Mitschke, A. Moreira, V. Amaral, J. Araujo. A model-driven traceability framework to software product line development. In Proceedings of the 4th European Conference on Model Driven Architecture - Traceability Workshop (ECMDA-TW'08). 2008. Berlin, Germany.

[16] S. Sendall, W. Kozaczynski, Model transformation: the heart and soul of model-driven software development. Software, IEEE, 2003. 20(5): p. 42-45.

[17] R. Oliveto. Traceability Management meets Information Retrieval Methods - Strengths and Limitations. In 12th European Conference on Software Maintenance and Reengineering, 2008. CSMR 2008. . 2008.

[18] A. Egyed. Resolving uncertainties during trace analysis. In Proceedings of the 12th ACM SIGSOFT twelfth international symposium on Foundations of software engineering. 2004. Newport Beach, CA, USA: ACM.

[19] J.H. Hayes, A. Dekhtyar, S.K. Sundaram, Advancing candidate link generation for requirements tracing: the study of methods. Software Engineering, IEEE Transactions on, 2006. 32(1): p. 4-19.

[20] P. Mäder, O. Gotel, I. Philippow. Enabling Automated Traceability Maintenance through the Upkeep of Traceability Relations. In Proceedings of the 5th European Conference on Model Driven Architecture - Foundations and Applications. 2009. Enschede, The Netherlands: Springer-Verlag.

[21] J. Bezivin, In search of a basic principle for model driven engineering. Novatica Journal, Special Issue, 2004.

[22] D.C. Schmidt, Model-driven engineering. IEEE Computer, 2006. 39(2): p. 25-31.

[23] P. Bernstein. Applying model management to classical meta data problems. In First Biennial Conference on Innovative Data Systems Research. 2003. Asilomar, CA, USA.

[24] L. Tratt, Model transformations and tool integration. Software and Systems Modeling, 2005. 4(2): p. 112-122.

[25] S. Beecham, N. Baddoo, T. Hall, H. Robinson, H. Sharp, Motivation in Software Engineering: A systematic literature review. Information and Software Technology, 2008. 50(9-10): p. 860-878.

[26] J. Biolchini, P.G. Mian, A.C.C. Natali, G.H. Travassos, Systematic review in software engineering. System Engineering and Computer Science Department COPPE/UFRJ, Technical Report ES, 2005. 679(05).

[27] F. Pino, F. García, M. Piattini, Software process improvement in small and medium software enterprises: a systematic review. Software Quality Journal, 2008. 16(2): p. 237-261.

[28] B. Kitchenham, S. Charters, Guidelines for performing Systematic Literature Reviews in Software Engineering, 2007.

[29] G.K. Olsen, J. Oldevik. Scenarios of traceability in model to text transformations. In Proceedings of the 3rd European Conference on Model Driven Architecture - Foundations and Applications (ECMDA-FA'07). 2007. Haifa, Israel: Springer-Verlag.

[30] J.R. Falleri, M. Huchard, C. Nebut. Towards a traceability framework for model transformations in kermeta. In Procedings of the 2nd European Conference on Model Driven Architecture - Traceability Workshop (ECMDA-TW'06). 2006. Bilbao, Spain.

[31] F. Jouault. Loosely coupled traceability for ATL. In 1st European Conference on Model-Driven Architecture Foundations and Applications (ECMDA-FA'05). 2005. Nuremberg, Germany.

[32] D. Steinberg, F. Budinsky, M. Paternostro, E. Merks, EMF: Eclipse Modeling Framework. 2nd Edition ed. The Eclipse Series, ed. E. Gamma, L. Nackman, and J. Wiegand2008: Addison-Wesley Professional.

[33] B. Grammel, S. Kastenholz. A generic traceability framework for facet-based traceability data extraction in model-driven software development. In Proceedings of the 6th ECMFA Traceability Workshop. 2010. Paris, France: ACM.

[34] M. Aleksy, T. Hildenbrand, C. Obergfell, M. Schwind. A pragmatic approach to traceability in model-driven development. In Process Innovation for Enterprise Software. 2009. Mannheim, Germany.
[35] N. Drivalos, D. Kolovos, R. Paige, K. Fernandes. A state-based approach to traceability maintenance. In Proceedings of the 6th ECMFA Traceability Workshop. 2010. Paris, France: ACM.
[36] N. Drivalos, D. Kolovos, R. Paige, K. Fernandes, Engineering a DSL for Software Traceability Software Language Engineering. In Software Language Engineering, 2009, Springer Berlin / Heidelberg. p. 151-167.
[37] D. Kolovos, R. Paige, F. Polack. On-demand merging of traceability links with models. In Procedings of the 2nd European Conference on Model Driven Architecture - Traceability Workshop (ECMDA-TW'06). 2006. Bilbao, Spain.
[38] R. Paige, N. Drivalos, D. Kolovos, K. Fernandes, C. Power, G. Olsen, S. Zschaler, Rigorous identification and encoding of trace-links in model-driven engineering. Software and Systems Modeling, 2011. 10(4): p. 469-487.
[39] N. Drivalos, R. Paige, K. Fernandes, D. Kolovos. Towards rigorously defined model-to-model traceability. In Proceedings of the 4th European Conference on Model Driven Architecture - Traceability Workshop (ECMDA-TW'08). 2008. Berlin, Germany.
[40] T. Levendovszky, D. Balasubramanian, K. Smyth, F. Shi, G. Karsai. A transformation instance-based approach to traceability. In Proceedings of the 6th ECMFA Traceability Workshop. 2010. Paris, France: ACM.
[41] I. Kurtev, K. Van den Berg, F. Jouault. Evaluation of rule-based modularization in model transformation languages illustrated with ATL. In Proceedings of the 2006 ACM symposium on Applied computing. 2006. Dijon, France: ACM.
[42] F. Jouault, B. Vanhooff, H. Bruneliere, G. Doux, Y. Berbers, J. Bezivin. Inter-DSL coordination support by combining megamodeling and model weaving. In Proceedings of the 2010 ACM Symposium on Applied Computing (SAC 2010). 2010. Sierre, Suiza: ACM.
[43] A. Jossic, M.D. Del Fabro, J.P. Lerat, J. Bezivin, F. Jouault. Model Integration with Model Weaving: a Case Study in System Architecture. In Systems Engineering and Modeling, 2007. ICSEM '07. International Conference on. 2007.
[44] F. Allilaire, Towards Traceability support in ATL with Obeo Traceability. Model Transformation with ATL, 2009: p. 150.
[45] M. Barbero, M.D. Del Fabro, J. Bezivin. Traceability and provenance issues in global model management. In Proceedings of International Conference on Systems Engineering and Modelling (ICSEM07). 2007.
[46] A. Boronat, J. Carsí, I. Ramos. Automatic Support for Traceability in a Generic Model Management Framework. In 1st European Conference on Model-Driven Architecture Foundations and Applications (ECMDA-FA'05). 2005. Nuremberg, Germany.
[47] J. von Pilgrim, B. Vanhooff, I. Schulz-Gerlach, Y. Berbers, Constructing and Visualizing Transformation Chains. In Model Driven Architecture - Foundations and Applications, I. Schieferdecker, A. Hartman, Editors, 2008, Springer Berlin / Heidelberg. p. 17-32.
[48] E. Guerra, J. de Lara, D. Kolovos, R. Paige, Inter-modelling: From Theory to Practice. In Model Driven Engineering Languages and Systems, D. Petriu, N. Rouquette, and ò. Haugen, Editors, 2010, Springer Berlin / Heidelberg. p. 376-391.
[49] P. Sánchez, D. Alonso, F. Rosique, B. Álvarez, A.P. Pastor, Introducing Safety Requirements Traceability Support in Model-Driven Development of Robotic Applications. IEEE Transactions on Computers, 2011. 60: p. 1059-1071.
[50] Y. Yu, Y. Lin, Z. Hu, S. Hidaka, H. Kato, Maintaining invariant traceability through bidirectional transformations for EMF, in Technical Report GRACE-TR2010, GRACE Center: Tokio.
[51] S. Melby, Traceability in Model Driven Engineering, 2007, Master Thesis. University of Oslo, Norway, http://urn.nb.no/URN:NBN:no-18721.

[52] J. Oldevik, T. Neple. Traceability in model to text transformations. In Procedings of the 2nd European Conference on Model Driven Architecture - Traceability Workshop (ECMDA-TW'06). 2006. Bilbao, Spain: Citeseer.
[53] B. Vanhooff, Y. Berbers. Supporting modular transformation units with precise transformation traceability metadata. In 1st European Conference on Model-Driven Architecture Foundations and Applications (ECMDA-FA'05). 2005. Nuremberg, Germany.
[54] S. Walderhaug, U. Johansen, E. Stav, J. Aagedal. Towards a generic solution for traceability in MDD. In 2nd European Conference on Model Driven Architecture - Traceability Workshop (ECMDA-TW'06). 2006. Bilbao, Spain.
[55] L. Bonde, P. Boulet, J.L. Dekeyser, Traceability and interoperability at different levels of abstraction in model-driven engineering. Applications of specification and design languages for SoCs: selected papers from FDL 2005, 2006: p. 263.
[56] K. Czarnecki, Domain Engineering. In Encyclopedia of Software Engineering, 2002, John Wiley & Sons, Inc.
[57] C. Atkinson, T. Kuhne, Model-Driven Development: A Metamodeling Foundation. IEEE Software, 2003. 20(5): p. 36-41.
[58] B. Selic, The pragmatics of model-driven development. Software, IEEE, 2003. 20(5): p. 19-25.
[59] K. Czarnecki, S. Helsen. Classification of Model Transformation Approaches. In OOPSLA'03 Workshop on Generative Techniques in the Context of Model-Driven Architecture. 2003.
[60] T. Margaria, B. Steffen, Continuous Model-Driven Engineering. IEEE Computer, 2009. 42(10): p. 106-109.
[61] M. Volter, From Programming to Modeling - and Back Again. Software, IEEE, 2011. 28(6): p. 20-25.
[62] A. Jiménez, J.M. Vara, V.A. Bollati, E. Marcos. Gestión de la trazabilidad en el desarrollo dirigido por modelos de Transformaciones de Modelos: una revisión de la literatura. In XVI Jornadas de Ingeniería del Software y Bases de Datos – JISBD 2011. 2011. A Coruña (España).
[63] I. Santiago, V. de Castro, J.M. Vara, E. Marcos. Evaluación de propuestas para la gestión de trazabilidad en el contexto de la Ingeniería Dirigida por Modelos. In XVI Jornadas de Ingeniería del Software y Bases de Datos – JISBD 2011. 2011. A Coruña (España).
[64] I. Galvao, A. Goknil. Survey of Traceability Approaches in Model-Driven Engineering. In Proceedings of the 11th IEEE International Enterprise Distributed Object Computing Conference. 2007. IEEE Computer Society.
[65] H. Schwarz, J. Ebert, A. Winter, Graph-based traceability: a comprehensive approach. Software and Systems Modeling, 2010. 9(4): p. 473-492.
[66] S. Winkler, J. von Pilgrim, A survey of traceability in requirements engineering and model-driven development. Software and Systems Modeling, 2010. 9(4): p. 529-565.