Research Review

A Systematic Literature Review on the Quality of UML Models

Marcela Genero, University of Castilla-La Mancha, Spain

Ana M. Fernández-Saez, University of Castilla-La Mancha, Spain

H. James Nelson, Southern Illinois University, USA

Geert Poels, Faculty of Economics and Business Administration, Ghent University, Belgium

Mario Piattini, University of Castilla-La Mancha, Spain

ABSTRACT

The quality of conceptual models directly affects the quality of the understanding of the application domain and the quality of the final software products that are ultimately based on them. This paper describes a systematic literature review (SLR) of peer-reviewed conference and journal articles published from 1997 through 2009 on the quality of conceptual models written in UML, undertaken to understand the state-of-the-art, and then identify any gaps in current research. Six digital libraries were searched, and 266 papers dealing specifically with the quality of UML models were identified and classified into five dimensions: type of model quality, type of evidence, type of research result, type of diagram, and research goal. The results indicate that most research focuses on semantic quality, with relatively little on semantic completeness; as such, this research examines new modeling methods vs. quality frameworks and metrics, as well as quality assurance vs. understanding quality issues. The results also indicate that more empirical research is needed to develop a theoretical understanding of conceptual model quality. The classification scheme developed in this paper can serve as a guide for both researchers and practitioners.

Keywords: Conceptual Model Quality, Conceptual Models, Software, Systematic Literature Review, Unified Modeling Language (UML)

INTRODUCTION

Software is becoming increasingly complex. So complex, in fact, that it is widely acknowledged that it is impossible to test every aspect of system software or application programs before release. One method of increasing understanding between the software developer and the customer, of deepening the understanding of how software works, and ultimately of reducing the complexity of software, is through the use of models (Thomas, 2004). Over the years, we have come to understand that modeling offers benefits to different stakeholders in software projects (Selic, 2003). In the very early stages of a project, models aid in understanding and communicating requirements. During development, architecture and design models guide

DOI: 10.4018/jdm.2011070103

the implementation of the system. Finally, models are used for test-generation (Offutt & Abdurazik, 1999) and for easing maintenance activities (Dzidek, Arisholm, & Briand, 2008).

Software development itself is becoming more model-centric (Mohagheghi, Dehlen, & Neple, 2009). The OMG Model Driven Architecture (OMG, 2003) and the recent growth of the Model-Driven Development (MDD) software engineering paradigm (Atkinson & Kühne, 2003) emphasizes the role of modeling in the development of software systems. MDD treats software development as a set of transformations between successive models from requirements to analysis, to design, to implementation, and to deployment (Thomas, 2004). MDD's defining characteristic is that software development's primary focus and products are models rather than computer programs.

The dominant question is no longer "Should we do modeling?" but "*How* should we do modeling?" This new focus on the modeling process, rather than on the software product resulting from the development activities, puts model quality in the forefront. There has been increasing interest, both in industry and academia, on methods and techniques for quality assessment, assurance, and improvement of models in software development and maintenance (Mohagheghi, Dehlen, & Neple, 2009). While there has been a great deal of research on software quality, there has been relatively little work on the quality of models, and the concept of model quality is poorly understood. Existing knowledge on software quality has limited applicability to model quality. Models have very different characteristics than source code: models have multiple views, may be used informally and casually rather than formally and with precision, can be used throughout all phases of the project, and so on.

In an effort to bring together the wide variety of modeling methods and forms, the *Unified Modeling Language* (UML) emerged in the 1990s as a standard modeling language for a wide spectrum of application domains (Rumbaugh, Booch, & Jacobson, 1998). This standardization has driven the advancement of modeling methods and tools and has enabled academics and practitioners alike to improve on the core structure of UML and engage in a healthy debate about the use, advancement, and basic beliefs about the modeling process. However, modeling research has tended to be more about improving UML to deal with special modeling cases than with improving model quality (Dobing & Parsons, 2006; Grossman, Aronson, & McCarthy, 2005).

In order to advance the field of conceptual modelling quality research, it is useful to explore the history of the field and to determine its current state of the art by locating, evaluating, and interpreting relevant research to date that is related to model quality with a focus on UML. This paper presents a systematic literature review (SLR) of papers dealing with the quality of UML models. A proper systematic literature review follows a rigorous and systematic approach, in particular that described by Brereton, Kitchenham, Budgen, Turner, and Khalil (2007), Kitchenham (2004), and Kitchenham and Charters (2007).

Six digital libraries containing thousands of academic research papers were searched, producing 266 papers dealing exclusively with UML model quality. These papers were classified in five dimensions (Table 3): quality type, type of evidence, research results, type of UML diagram, and research goals. These dimensions were chosen on the basis of previous research on model quality (though not necessarily restricted to UML) and conceptual frameworks that help define the concept of model quality. These dimensions are also useful to position new research activities appropriately, and the classified library of UML model quality research papers can serve as a valuable resource both for researchers and for practitioners.

An analysis of the papers shows the progress made in advancing UML model quality and identifies where gaps exist that could be areas for further investigation. The results indicate that there is no clear view of the real state of the field, although quality of models used in software development is a "hot topic" that needs further investigation (Wand & Weber, 2002).

Quality assurance techniques for software, such as testing, inspections, analysis, and measurement, are well established, but their application in the domain of UML models and MDD is still in an embryonic phase. However, starting in the year 2002 and continuing to the present, the topic has been well represented in international conferences such as ER (International Conference on Conceptual Modeling) and MODELS (International Conference on Model Driven Engineering Languages and Systems) and in various associated workshops, such as QiM (Quality in Modeling), MoDEVA (Model Validation), IWCMQ (International Workshop on Conceptual Modelling Quality), and QoIS (Quality of Information Systems).

The remainder of the paper is structured as follows. The next section presents a brief discussion of related work. This is followed by the SLR outline, a description of the activities of the SLR process, and a discussion of the results and their implications. The paper concludes with an analysis of the threats to validity of the results, the lessons learned with respect to performing the SLR, and finally a discussion of future research possibilities.

RELATED WORK

In related work, Moody (2005) performed a review of research on conceptual model quality using two literature search engines: Ingenta and Proquest. The objective of this review was to investigate the possible ways of structuring, developing, and empirically validating conceptual model quality frameworks. Moody's review identified forty approaches and considered the following issues: level of generality, scope, origin, and empirical validation. The study described some initial efforts towards developing a common standard for data model quality which may provide a model for future standardisation efforts.

Genero, Piattini, and Calero (2005a) performed an exploratory review of measures for UML class diagrams. This paper described the goal of each measure, indicating whether it has any theoretical or empirical validation, and indicated if there were any tools that support the measure's use. A total of nine metrics sets for UML class diagrams were identified. Some were defined specifically for UML class diagrams, while others were originally defined for measuring specific types of Object-Oriented design models, but could be tailored for use with UML class diagrams. The study revealed that even though a plethora of metrics exists for measuring different properties of UML class diagrams (such as size, cohesion, complexity, coupling, etc.) empirical evidence of their utility in practice is scarce. Moreover, the theoretical validation (i.e., the test that the metrics really measure the attribute they purport to measure following principles of Measurement Theory) was neglected in most of the proposals.

Pretorius and Budgen (2008) reviewed 33 papers published between 2001 and 2008 reporting empirical evidence on the use of UML models and forms. Using only the abstracts, they found that comprehension and metrics were the major topics for experimentation. The authors concluded that model quality and adoption experiences deserve further study.

Mohagheghi, Dehlen, and Neple (2009) reviewed 40 primary studies (including books, a PhD thesis, journals, workshops, proceedings, and published online) published between 2000 and 2007 and focused on model quality within the domain of model-driven and model-based software development. Their research examined quality goals, quality practices, and the types of models and modeling approaches discussed in the literature. The authors identified six quality goals in their literature: correctness, completeness, consistency, comprehensibility by humans and tools, confinement, and changeability. They also identified six quality practices, and that most models discussed were UML Models.

Lucas, Molina, and Toval (2009) reviewed 44 papers published between 2001 and 2007, focusing only on consistency within UML models. That is, across two or more UML diagrams that makes up a complete UML model. Their conclusion is that UML model consistency is a highly active and promising line of research, but that there are some important gaps in the literature. The authors address these gaps by introducing a formal consistency management language.

This SLR differs from the previous literature reviews presented above in three ways: a different goal, a more extensive and a more systematic review, and a more refined classification. The goal of our review is to identify "what has been done" and "what needs to be addressed in the future" in the context of quality of UML models. This contrasts with previous SLRs that have a rather narrow focus, such as measures for UML class diagrams, quality frameworks, empirical research, MDA, and consistency. The reviews presented in Genero, Piattini, and Calero (2005a) and Moody (2005) do not describe a systematic selection process, nor do they state clear criteria for inclusion or exclusion. Pretorius and Budgen (2008) present a systematic review but on a fairly small scale with only 33 papers considered. The review of Mohagheghi, Dehlen, and Neple (2009) used fewer digital resources, had only 40 primary studies, and the review process was not strictly systematic.

Our extensive literature review is based on a systematic search of six digital libraries, following the procedure described in Brereton, Kitchenham, Budgen, Turner, and Khalil (2007), Kitchenham (2004), and Kitchenham and Charters (2007) producing 266 papers for analysis. This study uses a more refined classification system that classifies each paper along five dimensions, producing a finer-grained classification. It identifies additional analysis possibilities, and may provide deeper insights into UML model quality and modelling quality in general.

SLR OUTLINE

The SLR research method consists of three activities: planning, execution, and reporting (Kitchenham & Charters, 2007). Each of these activities is divided into several steps. Planning includes dividing the workload amongst the researchers, determining how the researchers will interact and conduct the review, and developing the review protocol itself. The execution activity includes data retrieval, study selection, data extraction, and data synthesis. Finally, the reporting activity presents and interprets the results. The next sections contain a detailed description of the SLR activities.

PLANNING THE REVIEW

The planning activity includes defining the research questions and developing the search strategy, the inclusion / exclusion criteria, and the data extraction form.

Five research questions were proposed, based on previous research by Piattini, Genero, Poels, and Nelson (2005). The underlying motivation for the research questions was the goal of determining the amount of coverage of the UML model quality research area and these questions guided the design of the review process (Jørgensen & Shepperd, 2007). The research questions and the motivation for each are described in Table 1.

Although there are many collections of research papers available to choose from in both electronic and physical (paper) form, we limited the search to only electronic collections and considered only peer-reviewed journals, conferences, and workshops. While there is a great deal of additional conceptual modeling literature in books, working papers, web pages, magazine articles, white papers, and trade journals, the content of these sources has not been subjected to peer review and so their quality cannot be reliably determined.

We chose electronic collections that contain a wide variety of computer science and management information systems journals: SCOPUS database, Science@Direct with the subject Computer Science, Wiley InterScience with the subject of Computer Science, IEEE Digital Library, ACM Digital Library, and SPRINGER database. The search was restricted to only the first level. That is, the references of the selected papers were not searched to obtain more papers

Table 1. Research questions

Research questions	Main motivation
RQ1 Which types of UML model quality are investigated by researchers?	To discover the different types of model quality and specific quality characteristics that have been addressed by research.
RQ2 Which research methods are used in research on UML model quality?	To determine if the field has generally more applied or more basic research as well as to identify opportunities for future research.
RQ3 What is the nature of the research results on UML model quality?	To find the kind of outputs produced by UML model quality research and to assess the state of the field.
RQ4 What are the UML model quality research goals?	To determine where most of the research interest lies and which areas may be under-studied: exploring basic concepts, gathering knowledge of current practices, or aiming at advancing practice through design science.
RQ5 Which types of UML diagrams are the focus of the research on UML model quality?	To discover the UML diagrams that research has focused upon, to reveal the parts of UML that are considered more important than others, as well as to identify opportunities for further research.

on the subject. There are two reasons for this: relevant papers would have been found in the initial search process, and the references contain only the author and title, which may or may not indicate UML quality-related research.

The search terms used were constructed using the following steps (Brereton, Kitchenham, Budgen, Turner, & Khalil, 2007):

- Define the major terms.
- Identify alternative spellings, synonyms or related terms for major terms.
- Check the keywords in any relevant papers we already had.
- Use the Boolean OR to incorporate alternative spellings, synonyms or related terms.
- Use the Boolean AND to link the major terms.

The major search terms are “UML” and “Quality”. The alternative spellings, synonyms or terms related to the major terms are shown in Table 2.

Whenever a database or digital library did not allow the use of complex Boolean search strings, we designed different search strings for each of these databases and manually manipulated the searches in order to obtain the same results that had been achieved using the original search string. The search was performed on the full text of the article, except in those libraries that did not provide this capability. In that case, the search was restricted to the title and the abstract.

Papers were included that dealt with UML and the tangible results of the modelling process (the UML diagram), were written in English, and were published from 1997 through 2009. As UML was adopted by the Object Management Group in 1997 (OMG, 1997) it made no sense to search for papers before 1997. The final search was performed in March 2010, to allow as much time as possible for papers to appear in the digital libraries.

The following papers were excluded: pure discussion and opinion papers, studies available only in the form of abstracts or PowerPoint presentations, duplicates (for example, the same paper included in more than one database or in more than one journal), research focusing on issues other than UML model quality, or where quality is mentioned only as a general introductory term in the paper’s abstract, an approach or other type of proposal related to quality not being amongst the paper’s contributions. Papers were also excluded if they dealt with complexity of UML as a language (for example, how to make the UML language itself simpler) rather

Table 2. Search string

Major terms	Alternative terms
Quality	quality OR consistency OR maintainability OR understandability OR completeness OR comprehension OR comprehensibility OR testability OR defect OR effectiveness OR complexity OR readability OR metric OR measure OR efficiency OR validation OR verification OR layout
UML	UML OR Unified Modeling Language
Representation	Representation OR diagram OR model

than on the quality of the models produced by UML, and finally if the paper was a summary of a workshop presentation.

The extracted papers were tracked on a two-part form. The first part contained the general "demographic" information, such as title, authors, publication, year, and so on. The second part contained the multidimensional classification scheme. A set of five dimensions was used to classify the research, based on the research questions described above which, in turn, are based on work by Piattini, Genero, Poels, and Nelson (2005). This work has its foundations in Krogstie (1998), Lindland, Sindre, and Sølvberg (1994), and Nelson, Monarchi, and Nelson (2001). These dimensions are shown in Table 3 and a detailed description of the classification scheme is presented in the Appendix.

A similar classification scheme to the one used in this paper was employed in Poels, Nelson, Genero, and Piattini (2003) to categorize the papers published in the first edition of The International Workshop on Quality in Conceptual Modeling (IWQCM) held within the ER conference, and in Piattini, Genero, Poels, and Nelson (2005) to classify the chapters of the book "Quality of Software Conceptual Models" (Genero, Piattini, & Calero, 2005b). Matulevicius and Heymans (2007) and Matulevicius, Heymans, and Sindre (2006) used a similar classification scheme to evaluate modeling languages for contextualizing their research on conceptual modeling quality.

CONDUCTING THE REVIEW

The review process and timeline is shown in Table 4. Planning for the SLR began in July 2007 and papers dealing with UML model quality published between 1997 and the extraction date were retrieved in September 2007. Over 1500 papers were extracted, duplicates were removed, and data was analyzed over the next ten months. The title and abstract of each of the papers was examined and all papers not dealing with UML model quality research were excluded, reducing the total to 483. 144 duplicate papers were discarded, and the inclusion and exclusion criteria were then applied by reading the full text of each of the 339 remaining papers. Amongst the excluded papers were a number related to functional size measurement based on UML diagrams, as functional size is not related to model quality. Based on this analysis, the extraction and classification schemes were refined, primary studies were identified, follow-up studies were eliminated, and final classifications were made. The final 193 papers were then analyzed and the results were interpreted.

However, by this point, considerable time had passed, so a second phase was planned and executed. This phase began in March 2010 and included all UML quality papers published in 2008 and 2009. 979 additional papers were extracted, analyzed, and classified using the same process as described in the first phase. 74 papers remained after the second phase. Fi-

Table 3. Summary of the classification scheme

Dimensions	Categories
Type of Quality	Syntactic quality: correctness Semantic quality: consistency, completeness, correctness Pragmatic quality: maintainability, analyzability, understandability, testability, functionality, executability, reusability, complexity, dependability.
Type of Evidence / Research Method	Argumentation, example, experiment, case study, survey
Type of Research Result	Quality model, notation, method (technique, methodology, process, approach, or strategy).or algorithm, tool, metric, confirmation of knowledge, pattern, view, checklist (guideline, rule or modeling convention)
Research goal	Understanding, measuring, evaluating, assuring, improving
Type of Diagram	Structure diagrams, behavior diagrams interaction diagrams

nally, the results of the two selection phases were joined, resulting in 266 papers as primary studies to be analyzed in this SLR. The complete list of papers is available at http://alarcos.esi.uclm.es/SLR-QualityUMLModels

DATA SYNTHESIS AND RESULTS

For the purpose of the review analysis, i.e., addressing the research questions listed in Table 1, the 266 primary studies selected were classified according to the dimensions detailed in the classification scheme presented in Table 3. Based on this classification, the analysis reported in this section was aimed at finding answers to the research questions.

Model Quality (RQ1)

There are three main quality types: syntactic, semantic, and pragmatic (Lindland, Sindre, & Sølvberg, 1994; Unhelkar, 2005) with several subtypes defined under each type. The results of classifying the 266 research papers are shown in Table 5. Most of the research effort has concentrated on semantic quality (50.75%) followed by pragmatic quality (38.72%) with relatively little research effort on syntactic quality (5.64%). There are a few papers that cross quality types. Six papers deal with both syntactic and semantic quality, six deal with both semantic and pragmatic quality, and one addresses all three quality types.

Each quality type has associated with it a number of subtypes, described in Table 3. The count of research papers in each subtype is shown in Table 6. As with quality type, a paper may address more than one subtype, so the numbers will add up to more than 266, the actual number of papers in the study.

Semantic consistency is by far the semantic quality subtype that has been researched most. The papers that fall within this category investigate primarily issues of consistency that may arise when models are constructed using different types of UML diagrams. Next in line comes model correctness, whereas only 14 papers that deal with semantic quality address model completeness.

For pragmatic quality, understandability is a clear leader with maintainability coming in at a distant second place. Apart from complexity (11 papers), the other pragmatic quality subtypes are addressed in only a very few papers.

Research Method (RQ2)

There are many research methods to choose from when performing investigations into model quality. The results of the research method clas-

Table 4. Chronology of the development of activities in the SLR

Time	Planning	Conducting	Reporting	Outcomes
First phase				
July 2007	Protocol development			Review protocol.
Sept 2007		Data retrieval (until Sept 2007)		Form with the general information of the papers (1500 papers).
		Study selection upon abstracts and titles		Form with the general information of the selected papers (483 papers).
Mar 2008		Retrieval of the files of the primary studies		Repository of papers (483 papers).
Apr 2008		Remove duplicates		Form with the general information of the papers (399 papers).
Jul 2008	Protocol improvement	Pilot data extraction		Data extraction form with the classification scheme refined.
Aug 2008		Study selection and Data extraction upon the full text		Data extraction form completed with the classification of 215 primary studies.
Feb 2009		Resolution of doubts in classification of primary studies in group		Revisited data extraction form with classification of the primary studies **(193).**
Mar 2009		Data synthesis		
July 2009			Report the results of the SLR	Pilot report
Second phase				
Mar 2010		Update of searches Data retrieval (until Dec 2009)		Form with the general information of the papers (979).
Mar 2010		Study selection upon abstracts and titles		Form with the general information of the selected papers (140).
		Retrieval of the files of the primary studies		Repository of papers 140).
		Remove duplicates		Form with the general information of the papers (103).
Feb 2010		Study selection and Data extraction upon the full text		Data extraction form completed with the classification of primary studies (103)
Mar 2010		Resolution of doubts in classification of primary studies in group		Revisited data extraction with the classification of primary studies **(73)**
Apr 2010		Data synthesis		
Jul 2010			Report the results of the SLR	Final report

Table 5. Percentage of papers addressing different quality types

Type of quality	**Number**	**Percent**
Syntactic	15	5.64%
Semantic	135	50.75%
Pragmatic	103	38.72%
Syntactic + Semantic	6	2.26%
Syntactic + Pragmatic	0	0.00%
Semantic + Pragmatic	6	2.26%
Syntactic + Semantic + Pragmatic	1	0.38%
Total	**266**	**100.00%**

Table 6. Number of papers per quality characteristics

Syntactic	**Number**	
Correctness	21	100.00%
	Total	**21**
Semantic	**Number**	
Consistency	113	62.09%
Completeness	14	7.69%
Correctness	55	30.22%
	Total	**182**
Pragmatic	**Number**	
Maintainability	24	19.35%
Analyzability	1	0.81%
Understandability	78	62.90%
Testability	2	2.61%
Functionality	4	3.23%
Executability	2	1.61%
Reusability	1	0.81%
Complexity	11	8.87%
Dependability	1	0.81%
	Total	**124**

sification effort are shown in Table 7. Note that the total of the numbers (278) is higher than the total number of papers in the SLR (266). This is due to some papers falling into more than one category. For example, one paper is both a survey and a case study. In our classification, research methods can be empirical or non-empirical. The former category contains 29.86% of the papers and the latter 70.14%. The non-empirical category includes papers that use models only to illustrate the proposals made (e.g., methods, metrics, or guidelines).

Table 7. Number of papers per type of evidence

Research method	Number	Percent	Syntactic		Semantic		Pragmatic	
Empirical	**83**	**29.86%**	**2**	**9.09%**	**19**	**12.84%**	**62**	**57.41%**
Experiment	66	23.74%	2	9.09%	9	6.08%	55	50.93%
Case study	15	5.40%	0	0.00%	9	6.08%	6	5.56%
Survey	2	0.72%	0	0.00%	1	0.68%	1	0.93%
Non empirical	**195**	**70.14%**	**20**	**90.91%**	**129**	**87.16%**	**46**	**42.59%**
Speculation	26	9.35%	2	9.09%	19	12.84%	5	4.63%
Example	169	60.79%	18	81.82%	110	74.32%	41	37.96%
Literature Review	0	0.00%	0	0.00%	0	0.00%	0	0.00%
Total	**278**		**22**		**148**		**108**	

Although more than half of the papers in the SLR (60.79%) used examples to clarify the proposal made, there were relatively few empirical research papers, with experiments being the empirical research method that was used most often. Survey and literature review methods are (almost) absent in the selected papers. Breaking this down further (not shown in Table 7), 71.67% of these papers focused on the impact on model understandability of different modeling methods or styles. Experimental research tended to be carried out with Computer Science Students (72.22%) who are in their third, fourth or fifth year. Less frequent are subjects that are academic staff members (5.56%) or practitioners (22.22%). In addition, the use of students as subjects, as well as "toy" problems makes generalizability a serious concern. Combining this research question with model quality type (RQ1), we see that 55 of these papers used controlled experiments to test hypotheses on pragmatic quality.

Most of the papers in the SLR, by far, were non-empirical research on semantic quality. The most common of these was research that proposed model quality related modifications or that extended UML and demonstrated the problem and/or the utility of the research using one or more examples. Of the research on semantic quality that used examples, 70.27% of the papers focus on consistency issues.

Research Results (RQ3)

The papers were also classified by the kind of research output that was produced. The results of the SLR are shown in Table 8. Again, the total number is greater than 266, as some papers fall into more than one category. By far the most common research output is a new method. These methods can be quite varied, as we find methods for model validation, verification, transformation, and so on. The second most common paper, but far behind methods, is that which produces new knowledge. These are largely papers that employ empirical research methods (RQ2), as the testing of hypotheses can be seen as knowledge production.

Closely following Knowledge and in third place are papers that propose new tools: tools for automatic consistency checking between diagrams within a UML model, model-based checking tools, visualization tools, and so on. Metrics papers present a variety of metrics and techniques for measuring different model characteristics, such as size, complexity, consistency, and so on. The fifth most common type of paper presents rules, modeling conventions, guidelines and checklists. Other types of research results are scarce.

If we focus on the five categories noted, we can see that the proposal of methods, tools, and rules relates mostly to semantic quality,

Table 8. Number of papers per type of research result

Type of Result	Number	Percent
Formal semantics	3	1.01%
Framework	3	1.01%
Knowledge	55	18.46%
Method	119	39.93%
Metrics	28	9.40%
Notation	10	3.36%
Pattern	4	1.34%
Quality model	1	0.34%
Tool	50	16.78%
View	3	1.01%
Checklist, rules, modeling conventions, and guidelines	22	7.38%
Total	**298**	**100.0%**

whereas the knowledge and metrics research output relates mostly to the pragmatic quality. For the knowledge-producing studies, this result is consistent with the finding that these employ mostly experiments which focus on pragmatic quality and specifically on understandability.

Of the methods that deal with semantic quality issues, we can see that the majority are approaches to improve the consistency of UML diagrams. The same observation holds for tools that are proposed for semantic quality issues. Most of them focus on consistency, although a substantial percentage of the tools proposed in the papers relate to semantic correctness. In addition, most of the rules, modeling conventions, guidelines and checklists are related to semantic quality, especially to consistency.

Most pragmatic quality metrics papers propose metrics for assessing or predicting the maintainability of UML models, while the next largest percentage focuses on measuring understandability. These two categories are closely related; before a diagram can be modified, it must be understood. It is noteworthy that 76% of the papers in these two categories include a validation of the metrics through one or more controlled experiments or a case study in addition to the metrics definition (Table 9).

Research Goals (RQ4)

The purpose of investigating the goals of the research papers is to determine where UML model quality research interest lies and to determine which areas may be under-studied. As shown in Table 10, there are 121 papers (which represents 45.49% of the total) related to assuring quality, 85 papers (31.95%) are related to evaluating quality, and 38 papers (14.29%) to measuring quality. The other two categories, improving and understanding, together account for less than 9% of the papers.

Research on applied areas of UML quality assurance techniques and the evaluation of UML quality account for more than three-quarters (77.44%) of the papers published. This is not surprising, as quality assurance is a critically important topic. The other basic research topics are important for advancing the state of the art of UML model quality but they are much less well-represented in the survey. This can mean that a given topic is under-

Table 9. Crossing type of result with type of quality and quality characteristic

	Method	Knowledge	Tool	Metrics	Rule, modeling convention, checklist, guideline
Pragmatic	**18.25%**	**76.06%**	**22.03%**	**91.18%**	**24.0%**
Dependability	0.73%	0.00%	0.00%	0.00%	0.0%
Executability	0.73%	0.00%	3.39%	0.00%	0.0%
Functionality	1.46%	2.82%	0.00%	2.94%	0.0%
Maintainability	3.65%	9.86%	3.39%	26.47%	0.0%
Reusability	0.73%	0.00%	0.00%	0.00%	0.0%
Complexity	0.00%	1.41%	1.69%	23.53%	4.0%
Testability	0.00%	0.00%	1.69%	2.94%	0.0%
Understandability	10.95%	60.56%	11.86%	35.29%	20.0%
Analyzability	0.00%	1.41%	0.00%	0.00%	0.0%
Semantic	**74.45%**	**19.72%**	**62.71%**	**8.82%**	**72.0%**
Completeness	4.38%	7.04%	3.39%	0.00%	8.0%
Consistency	55.47%	9.86%	38.98%	5.88%	48.0%
Correctness	14.60%	2.82%	20.34%	2.94%	16.0%
Syntactic	**7.30%**	**4.23%**	**15.25%**	**0.00%**	**4.0%**
Correctness	7.30%	4.23%	15.25%	0.00%	4.0%

Table 10. Results of papers per research goal

Research Goal	Number	Percent
Improving	15	5.64%
Assuring	122	45.49%
Measuring	38	14.29%
Evaluating	85	31.95%
Understanding	7	2.63%
Total	**266**	**100.0%**

studied or that it has yet to find acceptance with journals, or it may be that both of these states are the case for the topic in question.

UML Diagram (RQ5)

While over 65% of the papers in the survey focused on the quality of a specific kind of UML diagram, nearly 30% examined UML diagrams as a whole. The original 1997 version of UML had nine different kinds of diagrams that allowed systems to be modelled from many different viewpoints. UML 2.0 introduced four new diagrams, making a total of 13 diagrams. One of these, the communication diagram, was renamed from the original UML collaboration diagram. We use the name of the original version, as it is more widespread.

The type of diagram that has been studied most is the class diagram, followed by

Table 11. Number of papers per type of diagram

Type of diagram	Number	Percent
Class diagrams	83	25.30%
Sequence diagrams	34	10.37%
Activity diagrams	15	4.57%
Use case diagrams	21	6.40%
Statechart diagrams	55	16.77%
Collaboration diagrams	8	2.44%
Component diagrams	3	0.91%
Object diagrams	2	0.61%
Package diagrams	3	0.91%
Deployment diagrams	1	0.30%
No specific diagram	103	31.40%
UML 2.0 new diagrams	0	0.0%
Total	**328**	**100.0%**

statechart diagrams and sequence diagrams. Research on UML model quality has placed much less attention on use case diagrams and activity diagrams. Very few papers found take as their focus collaboration, component or package diagrams. This is an interesting result, as class diagrams, state-transition diagrams, and sequence diagrams have had a long history before the introduction of UML. Use case diagrams, for example, are relatively newer. One would expect that older diagrams would have less research and that more research would be required on understanding and on improving the quality of the newer diagrams. The literature review found no references to any of the four new diagrams of UML 2.0. These results can be seen in Table 11.

Additional Results

Beyond the investigation into the state of research into the quality of UML models, it is useful for producers and consumers of research to be aware of the various outlets for the research and the growth of the field.

As shown in Figure 1, there is a clear progression in the number of publications that appear each year. This figure may show that interest in this subject has been growing over time, reaching its highest point in 2007.

When analysing the types of publication, we found that 63.53% of the papers (169 papers) were published in conferences, 22.93% in journals (61 papers) and 13.53% in workshops (36 papers). Without a doubt, the quality of UML models has been considered a “hot topic”, given the number of publications dealing with it.

Table 12 shows the publications with the largest number of papers on UML model quality. The first three, the International Conference on Model Driven Engineering Languages and Systems (MODELS), which formerly was called UML, Electronic Notes in Theoretical Computer Science, and Information and Software Technology have 16, 15 and 9 papers each, together representing 15.04% of the total. The next one, the International Conference on Software Engineering (ICSE), has 8 papers, representing 3.01% of the total.

DISCUSSION

This systematic literature review discovered 266 papers in peer-reviewed journals, conferences,

Figure 1. Number of papers per year

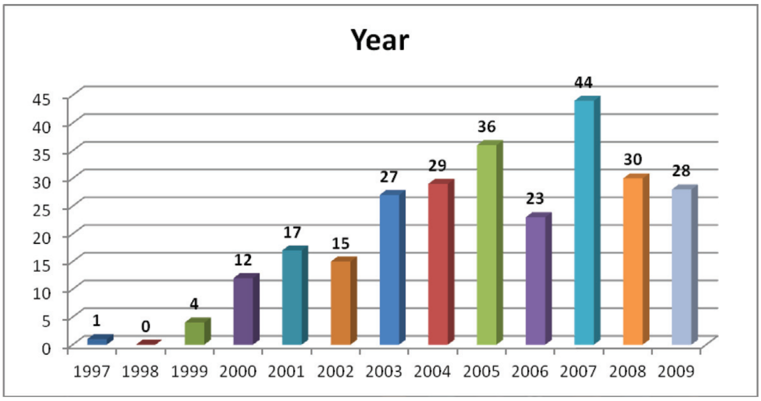

Table 12. Number of papers per type of publication

Publication	Number	Percent
International Conference on Model Driven Engineering Languages and Systems (MODELS, formerly UML)	16	6.04%
Electronic Notes in Theoretical Computer Science	15	5.66%
Information and Software Technology	9	3.40%
International Conference on Software Engineering (ICSE)	8	3.02%
ER Workshops	7	2.64%
ACM SIGSOFT Software Engineering Notes	6	2.26%
Journal of Systems and Software	6	2.26%
ACM symposium on software visualization (SoftVis)	5	1.89%
Empirical Software Engineering	5	1.89%
International Conference on Automated software engineering (ASE)	5	1.89%
Asia-Pacific Software Engineering Conference (APSEC)	4	1.51%
Australian Software Engineering Conference (ASWEC)	4	1.51%

and workshops and classified them into five dimensions represented by the five research questions presented above. In this section we discuss the results and draw implications from the classifications.

Research Question 1 asked, “Which types of UML model quality have been investigated by researchers?” The results show a clear ordering which may indicate the relative importance that researchers attach to each quality type. The order is: (1) semantic quality (i.e., correctness and completeness of the model with regard to the system to be modelled), (2) pragmatic quality (i.e., aspects relating to the use of the model), and (3) syntactic quality (i.e., syntactical correctness of the model). Whereas it is difficult to explain why semantic quality has received more research attention than pragmatic quality, except perhaps for the focus on model consistency found in the papers (113 out of 266 papers), which we try to interpret below, less than ten percent of the papers address syntactic quality, but this result is not surprising. That is because most, if not all, modelling tools enforce syntactic correctness automatically, so that if a particular element is not syntactically correct, the tool will not allow it to be placed.

Semantic quality research has mainly focused upon consistency issues. A plausible reason for the attention paid to consistency issues is that UML offers 13 types of diagrams, some of which have overlapping semantics and purpose. Moreover, little guidance is offered on when to use these different diagram types. It seems logical that researchers have investigated ways to cope with possible inconsistencies that may arise when multiple diagrams (of different types) are used to model the same system, from different (though sometimes related, even overlapping) points of view. Research on semantic quality has paid less attention to issues of semantic correctness (i.e., is the system correctly modelled?), and especially semantic completeness (i.e., are all relevant elements of the system modelled?). This result may be due to the considerable difficulty of proposing new approaches to ensuring that a model is complete and correct when compared against a domain, raw descriptions of system requirements or process structure and flows, and so on. While ensuring inter-diagram consistency is an important topic for UML model quality research, it can be argued that without proper attention for model completeness and correctness, consistency will not help in building a "good" information system.

With respect to pragmatic quality, research has emphasized the understandability, and, to a lesser extent, the maintainability of UML models. There are several possible reasons for this. Firstly, these quality characteristics can relatively easily be operationalized in research, compared to other pragmatic quality characteristics like functionality, complexity, and reusability; for example through comprehension questions and modification tasks. Secondly, the research on pragmatic quality issues has focused mainly on the use of UML models to facilitate the communication between stakeholders. Models are thus seen as instruments carrying information over from one party to another (which means that they need to be easy to understand). This observation can be caused 'by construction', as originally pragmatic quality was defined by Lindland, Sindre, and Sølvberg (1994) and Unhelkar (2005) as being "the extent to which a model is understood." Once understood, modifications can be made, so the next logical step is to investigate the maintainability of models.

Research Question 2 asked, "Which research methods are used in research on UML model quality?" The finding that only 29.86% of the papers back up their claims with an empirical study is remarkable. Most of the other papers do employ one or more examples to illustrate the problem researched and the solution proposed, but these examples cannot be seen as evaluations of the proposals made. A plausible reason for the low presence of empirical studies is that the SLR also looked at conference and workshop papers, for which demands of completeness of the research are less stringent than with journal papers. It is common that in conference and workshop papers (which must often adhere to strict limits on length) only a problem statement is given, followed by the development of a solution, but that the empirical evaluation of the solution is mentioned as future work. Most of the papers that present an empirical evaluation are studies employing experiments. The use of other methods (surveys, literature review) is rare. Studies aimed at evaluating proposals made with regard to quality issues are an obvious opportunity for further research. An increasing number of such studies would also indicate that the field is maturing. Good case-study research is another opportunity, as yet seldom exploited by research on UML model quality.

Research Question 3 asked, "What is the nature of the research results on UML model quality?" Nearly half the reviewed papers propose a method related to some UML model quality issue or action, mostly with respect to semantic quality, and model consistency in particular. Other typical research outputs include (in decreasing order of frequency) tools, metrics, and what could be described as instruments providing guidance to modelers (rules, modeling conventions, guidelines and checklists). Tools and instruments also largely aim at semantic quality, while most metrics measure pragmatic quality characteristics related to maintainability and understandability. A minority of papers, less than one out of five, had as a goal to increase or confirm existing knowledge on UML model quality. These papers mostly report on experiments that evaluate the effectiveness or efficiency of research outputs related to pragmatic quality, in particular understandability, or they aim at validating metrics and finding relationships between UML model quality variables.

We believe that these results are an indication that research on UML model quality is primarily conducted in a design science tradition, focusing on the development of new artifacts that advance the state-of-the-practice and aim at providing solutions to real-world problems experienced by modelers (e.g., the problem of how to ensure the consistency between different UML diagrams that compose a model). The lack of evaluation studies for these artifacts could be a consequence of the decision to also include conference and workshop papers in the SLR, as noted before. Nevertheless, if such evaluation is not present (i.e., not taken up in the complete research publications in journal papers), then this would indicate an incomplete design science research cycle, which could be interpreted as an indication of immaturity. The lack of 'knowledge gathering' studies may then indicate a premature focus on providing instruments to deal with quality issues, without a profound knowledge of the nature of UML model quality and its influencing factors.

Research Question 4 asked, "What are the UML model quality research goals?" The most important goal is quality assurance, which examines how to ensure that the modelling process actually produces a quality model. This is followed closely by quality evaluation, which compares quality measurements against real world experiences. This is most likely related to a tendency to propose new methods, tools and other quality instruments (RQ3) via non-empirical research methods (RQ2).

Only three percent of the papers had as a goal increasing knowledge about model quality. This follows the results of RQ3 where the research results also had relatively few papers regarding knowledge. The goal of understanding appears to be of little direct importance, though measuring and evaluating are important and may also help to acquire a better understanding.

Research Question 5 asked, "Which types of UML diagrams are the focus of the research on UML model quality?" Forty percent of the reviewed papers did not look at any UML diagram in particular, so the research presented in these papers is on a general level. If specific diagram types were targeted, these were then, in order of frequency, structure diagrams (almost exclusively class diagrams), behavior diagrams (mainly statechart diagrams) and interaction diagrams (mainly sequence diagrams). The types of diagram that were newly introduced in UML 2.0 have not been investigated yet from a quality perspective, and interaction diagrams (for example, collaboration diagrams) have received very little research attention.

These results reflect how frequent the various diagrams are used in practice, with one significant difference. Research indicates that the diagrams that are used most in modelling software systems are, in order of decreasing frequency: use case diagrams, class diagrams, sequence diagrams, and statechart diagrams (Dobing & Parsons, 2006; Erickson & Siau, 2007; Grossman, Aronson, & McCarthy, 2005). Erickson and Siau (2007) conclude that the most important UML diagrams are: class, use cases, sequence and statechart diagrams, and they should comprise a UML kernel. Significantly, the diagram that is used most in practice, the use case diagram, has received little attention from UML model quality research. A possible reason for this is that model quality research in general has mainly investigated structural diagrams or data models, and much less attention has been given to models that represent system behavior and interaction (Recker, Rosemann, & Krogstie, 2007).

Our results parallel the results of Moody (2005). Moody's review of forty papers identified twelve major theoretical and practical issues in existing research: proliferation of proposals, different levels of generality, lack of empirical testing, of adoption in practice, of agreement on concepts and terminology, of consistency with related fields and standards, of measurement, of evaluation procedures, of guidelines for improvement, of knowledge about practices, and a focus on static models and on product quality. Our results indicate that the issues identified by Moody for conceptual modelling quality research in general have applicability to UML quality research in particular.

THREATS TO VALIDITY

The main threats to the validity of a SLR are publication selection bias, inaccuracy in data extraction, and misclassification (Sjøberg et al., 2005). We acknowledge that it is impossible to achieve complete coverage of everything written on a topic. We used six digital sources, including journals, conferences and workshops which are relevant to software engineering. The scope of journals and conferences covered in this SLR is sufficiently wide to attain reasonable completeness in the field studied. We did not include additional papers such as grey literature (technical reports, books, etc.) as these tend to be secondary sources. Most grey literature either has its source in peer-reviewed papers or will become peer-reviewed papers, or both conditions may be true for a given piece of work. Some relevant papers may therefore have not been included, but our knowledge of the subject leads us to believe that there are not many such cases.

To help ensure an unbiased selection process, we defined research questions in advance, organized the selection of articles as a multistage process, involved five researchers in this process, and documented the reasons for inclusion/exclusion as suggested in Liu, Dehlinger, and Lutz (2007). As was discussed above, the decisions to select the papers to be included as primary studies in this SLR were made by multiple researchers and the process followed rigorous rules. A further challenge was that there is no keyword standard that we are aware of which distinguishes between different quality characteristics, or methods in empirical software engineering that could be used to extract quality characteristics and research methods in a consistent manner.

Moreover, article duplication is a potential threat to frequency counts and the statistics in this SLR. The structure of the database is designed to handle duplication, but one threat would be that of duplication going undetected. However, at least two people have read through all the relevant articles and have not detected any further duplicates.

The data was extracted from the papers by one researcher (the first author of the paper) and checked by the second author. When necessary, disagreements were resolved through discussion, involving the rest of authors. Data extraction and classification from prose is difficult at the outset; the lack of standard terminology and standards for reporting empirical studies and for defining quality characteristics in software engineering may have resulted in some inaccuracies in the data extraction and this may have resulted in a misclassification. However, we believe that the extraction and selection process was rigorous and that it followed the guidelines provided in Brereton, Kitchenham, Budgen, Turner, and Khalil (2007), Kitchenham (2004), and Kitchenham and Charters (2007). We also judge that the use of multiple experts performing the classification reduced the risk of misclassification. The classification scheme that we provide in this paper may be used as a starting point by future researchers.

LESSONS LEARNED WITH RESPECT TO PERFORMING THE SLR

It is usually not possible to judge the relevance of a study from a review of the abstract alone. The standard of IT and software engineering abstracts is too poor to rely on when selecting primary studies; it is therefore necessary to review the full text. When used properly, structured abstracts are very useful for improving the quality and usefulness of the abstract. Structured abstracts must contain the following sections: 1) Context (the importance and relevance of the research), 2) Objectives (the main objectives pursued), 3) Methods (the research method followed and the proposal provided to attain the objectives), and 4) Results (the main findings and conclusions obtained).

The search string is extremely long. Due to the limitation of the search engines, we observed that such a long string could not be searched directly. It was therefore necessary to tailor the search string to each digital library by splitting the original and combining the results manually. Current search engines are not designed to support systematic literature reviews. Unlike medical researchers, software engineering researchers need to perform resource-dependent searches (Brereton, Kitchenham, Budgen, Turner, & Khalil, 2007).

CONCLUSIONS AND FUTURE WORK

This paper reviews UML model quality papers published in journals, conferences and workshops found in six digital libraries and tries to support other researchers and practitioners through a library of papers on UML model quality which have been classified according to the following dimensions: type of quality, context of study, type of diagram, type of research result, research method and research goal, based on Piattini, Genero, Poels, and Nelson (2005). The SLR was carried out following the guidelines in Brereton, Kitchenham, Budgen, Turner, and Khalil (2007), Kitchenham (2004), and Kitchenham and Charters (2007) making this study both rigorous and fair.

Only 29.86% of the proposals collected carried out some kind of empirical validation. This fact reveals the need for further validation, i.e., replications made by other researchers different from the ones that make the proposals. In addition, we encourage experimental material to be available for encouraging replication. The repository of models proposed in the context of the MiSE workshops will contribute to carrying out empirical studies, proving UML models taken from real projects (France, Bieman, & Cheng, 2007).

Based on our (to some extent subjective) interpretation of the review results, we have several recommendations. First of all, that much more effort be spent on empirical research into conceptual model quality in general and UML quality in particular. There is a proliferation of tools and extensions for UML, but little indication that these tools and extensions really improve the quality of UML models. A coordinated effort of empirical research, the use of meta-analysis for integrating experiments, and experimentation using diagrams from real-world projects is needed in order to build a solid body of knowledge. Secondly, more interaction is needed between academia and industry. Somewhat paralleling the lack of empirical evidence, academic research does seem to be "ivory tower" research, with little input from real-world problems and issues. Problems, diagrams, and projects need to inform research, and basic research needs to be able to be easily applied. Thirdly, UML model quality research seems to concentrate on three types of quality (syntactic, semantic, pragmatic), yet there is no consensus on the quality characteristics addressed nor on their definitions. Finally, the topic needs to mature, with many more peer-reviewed articles published in leading journals.

Conceptual modelling quality is an important topic, with academia and industry both recognizing that it is critical to "get the model right." If the model is not correct and complete, it will be very difficult for the software systems based on that model to be correct and complete as well. We hope that this paper can serve as a guide to the contributions from past research in the area, as well as being useful as a foundation for future research.

ACKNOWLEDGMENT

This research is part of the following projects: MEDUSAS (CDTI-MICINN and FEDER IDI-20090557), ORIGIN (CDTI-MICINN and FEDER IDI-2010043(1-5), PEGASO/MAGO (MICINN and FEDER, TIN2009-13718-C02-01), EECCOO (MICINN TRA2009_0074), MECCA (JCMM PII2I09-0075-8394) and IMPACTUM (JCCM PEII11-0330-4414).

REFERENCES

Atkinson, C., & Kühne, T. (2003). Model-driven development: A metamodeling foundation. *IEEE Software*, *20*(5), 36–41. doi:10.1109/MS.2003.1231149

Brereton, P., Kitchenham, B., Budgen, D., Turner, M., & Khalil, M. (2007). Lessons from applying the systematic literature review process within the software engineering domain. *Journal of Systems and Software*, *80*(4), 571–583. doi:10.1016/j.jss.2006.07.009

Dobing, B., & Parsons, J. (2006). How UML is used. *Communications of the ACM*, *49*(5), 109–113. doi:10.1145/1125944.1125949

Dzidek, W. J., Arisholm, E., & Briand, L. C. (2008). A realistic empirical evaluation of the costs and benefits of UML in software maintenance. *IEEE Transactions on Software Engineering*, *34*(3), 407–432. doi:10.1109/TSE.2008.15

Erickson, J., & Siau, K. (2007). Theoretical and practical complexity of modeling methods. *Communications of the ACM*, *50*(8), 46–51. doi:10.1145/1278201.1278205

Fenton, N., Pfleeger, S. L., & Glass, R. L. (1994). Science and Substance: A Challenge to Software Engineers. *IEEE Software*, *11*(4), 86–95. doi:10.1109/52.300094

France, R., Bieman, J., & Cheng, B. (2007). Repository for model driven development (ReMoDD). In *Proceedings of the MoDELS Workshops* (pp. 311-317).

Genero, M., Piattini, M., & Calero, C. (2005a). A survey of metrics for UML class diagrams. *Journal of Object Technology*, *4*(9), 59–92. doi:10.5381/jot.2005.4.9.a1

Genero, M., Piattini, M., & Calero, C. (2005b). *Metrics for software conceptual models*. London, UK: Imperial College Press. doi:10.1142/9781860946066

Grossman, M., Aronson, J. E., & McCarthy, R. V. (2005). Does UML make the grade? Insights from the software development community. *Information and Software Technology*, *47*(6), 383–397. doi:10.1016/j.infsof.2004.09.005

International Organization for Standardization. (1998). *ISO/IEC 9126: Information Technology - Software product quality*. Geneva, Switzerland: International Organization for Standardization.

Jørgensen, M., & Shepperd, M. J. (2007). A systematic review of software development cost estimation studies. *IEEE Transactions on Software Engineering*, *33*(1), 33–53. doi:10.1109/TSE.2007.256943

Kitchenham, B. (2004). *Procedures for performing systematic reviews* (Tech. Rep. No. TR/SE-0401). Staffordshire, UK: Keele University.

Kitchenham, B., & Charters, S. (2007). *Guidelines for performing systematic literature reviews in software engineering*. Staffordshire, UK: Keele University.

Krogstie, J. (1998). Integrating the understanding of quality in requirements specification and conceptual modeling. *ACM SIGSOFT Software Engineering Notes*, *23*(1), 86–91. doi:10.1145/272263.272285

Lindland, O. I., Sindre, G., & Sølvberg, A. (1994). Understanding quality in conceptual modeling. *IEEE Software*, *11*(2), 42–49, 267. doi:10.1109/52.268955

Liu, J., Dehlinger, J., & Lutz, R. (2007). Safety analysis of software product lines using state-based modeling. *Journal of Systems and Software*, *80*(11), 1879–1892. doi:10.1016/j.jss.2007.01.047

Lucas, F. J., Molina, F., & Toval, A. (2009). A systematic review of UML model consistency management. *Information and Software Technology*, *51*(12), 1631–1645. doi:10.1016/j.infsof.2009.04.009

Matulevicius, R., & Heymans, P. (2007). Comparing goal modelling languages: An experiment. In *Proceedings of the Conference on Requirements Engineering: Foundation for Software Quality* (pp. 18-32).

Matulevicius, R., Heymans, P., & Sindre, G. (2006). Comparing goal-modelling tools with the re-tool evaluation approach. *Information Technology and Control*, *35*(3A), 276–284.

Mohagheghi, P., Dehlen, V., & Neple, T. (2009). Definitions and approaches to model quality in model-based software development - A review of literature. *Information and Software Technology*, *51*(12), 1646–1669. doi:10.1016/j.infsof.2009.04.004

Moody, D. L. (2005). Theoretical and practical issues in evaluating the quality of conceptual models: current state and future directions. *Data & Knowledge Engineering*, *55*(3), 243–276. doi:10.1016/j.datak.2004.12.005

Nelson, H., Monarchi, D., & Nelson, K. (2001). Ensuring the "goodness" of a conceptual representation. In *Proceedings of the 4th European Conference on Software Measurement and ICT Control*, Heidelberg, Germany.

Neto, A. D., Subramanyan, R., Vieira, M., Travassos, G. H., & Shull, F. (2008). Improving Evidence about Software Technologies: A Look at Model-Based Testing. *IEEE Software*, *25*(3), 10–13. doi:10.1109/MS.2008.64

Offutt, J., & Abdurazik, A. (1999). Generating tests from UML specifications. In R. France & B. Rumpe (Eds.), *Proceedings of the Conference on Unified Modeling Language* (LNCS 1723, pp. 416-429).

OMG. (1997). *Object Management Group - UML.* Retrieved from http://www.uml.org/

OMG. (2003). *MDA guide (Vol. version 1.0.1).* Retrieved from http://www.omg.org/docs/omg/03-06-01.pdf.

OMG. (2005). *The Unified Modeling Language. Documents associated with UML Version 2.0.* Retrieved from http://www.omg.org/spec/UML/2.0

Piattini, M., Genero, M., Poels, G., & Nelson, J. (2005). Towards a framework for conceptual modelling quality . In Genero, M., Piattini, M., & Calero, C. (Eds.), *Metrics For software conceptual models* (pp. 1–18). London, UK: Imperial College Press. doi:10.1142/9781860946066_0001

Poels, G., Nelson, J., Genero, M., & Piattini, M. (2003). Quality in conceptual modeling - New research directions. In A. Olivé, M. Yoshikawa, & E. S. K. Yu (Eds.), *Proceedings of the Conference on Advanced Conceptual Modeling Techniques* (LNCS 2784, pp. 243-250).

Pretorius, R., & Budgen, D. (2008). A mapping study on empirical evidence related to the models and forms used in the UML. In *Proceedings of the 2nd ACM-IEEE International Symposium on Empirical Software Engineering and Measurement*, Kaiserslautern, Germany (pp. 342-344).

Recker, J., Rosemann, M., & Krogstie, J. (2007). Ontology- versus pattern-based evaluation of process modeling languages: A comparison. *Communications of the Association for Information Systems*, *20*(48), 774–799.

Rumbaugh, J., Booch, G., & Jacobson, I. (1998). *Unified modeling language reference manual*. Reading, MA: Addison-Wesley.

Selic, B. (2003). The pragmatics of model-driven development. *IEEE Software*, *20*, 19–25. doi:10.1109/MS.2003.1231146

Shull, F., Singer, J., & Sjøberg, D. I. K. (2008). *Guide to Advanced Empirical Software Engineering*. Berlin, Germany: Springer. doi:10.1007/978-1-84800-044-5

Sjøberg, D. I., Hannay, J. E., Hansen, O., Kampenes, V. B., Karahasanovic, A., Liborg, N., & Rekdal, A. C. (2005). A survey of controlled experiments in software engineering. *IEEE Transactions on Software Engineering*, *31*, 733–753. doi:10.1109/TSE.2005.97

Thomas, D. (2004). MDA: Revenge of the modelers or UML utopia? *IEEE Software*, *21*, 15–17. doi:10.1109/MS.2004.1293067

Unhelkar, B. (2005). *Verification and validation for quality of UML 2.0 models*. New York, NY: Wiley Interscience. doi:10.1002/0471734322

Wand, Y., & Weber, R. (2002). Research commentary: Information systems and conceptual modeling - A research agenda. *Information Systems Research*, *13*(4), 363–376. doi:10.1287/isre.13.4.363.69

Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B., & Wesslén, A. (2000). *Experimentation in Software Engineering: An Introduction*. Norwell, MA: Kluwer Academic Publishers.

Zelkowitz, M., & Wallace, D. (1997). Experimental validation in software engineering. *Information and Software Technology*, *39*, 735–743. doi:10.1016/S0950-5849(97)00025-6

Zelkowitz, M., Wallace, D., & Binkley, D. W. (2003). Experimental validation of new software technology. In *Lecture Notes on Empirical Software Engineering* (pp. 229–263). Singapore: World Scientific Publishing. doi:10.1142/9789812795588_0006

*Marcela Genero is Associate Professor in the Department of Technologies and Information Systems at the University of Castilla-La Mancha, Ciudad Real, Spain. She received her MSc. degree in Computer Science in the Department of Computer Science of the University of the South, Argentina, in 1989, and her Ph.D. at the University of Castilla-La Mancha, Ciudad Real, Spain in 2002. She has published in prestigious journals (*Information and Software Technology, Journal of Software Maintenance and Evolution: Research and Practice, Data and Knowledge Engineering, Empirical Software Engineering, European Journal of Information Systems*, etc.). Along with Mario Piattini and Coral Calero she edited the books entitled “*Data and Information Quality*” (Kluwer, 2001), and “*Metrics for Software Conceptual Models*” (Imperial College, 2005). She is a member of the International Software Engineering Research Network (ISERN). Her research interests are: empirical software engineering, software metrics, conceptual models quality, quality in model-driven development, etc.*

Ana M. Fernández-Sáez has a MSc in Computer Science from the University of Castilla-La Mancha, Ciudad Real, Spain (2009). She is member of the Alarcos research group and Ph.D student at the Department of Technologies and Information Systems at the same university. Her research interests include: UML model quality, quality in model-driven development, software measures and empirical software engineering.

Jim Nelson is an assistant professor of Management Information Systems at Southern Illinois University, Carbondale. He received his BS in Computer Science from California Polytechnic State University, San Luis Obispo, and his MS and PhD in Information Systems from the University of Colorado, Boulder. His research interests include the quality of conceptual models, investigating how people make IT paradigm shifts, and determining the business value of information technology. Dr. Nelson generally teaches the more technical courses in information systems including object oriented technology, systems analysis and design, database theory and practice, and business data communications.

Geert Poels is a professor with the rank of senior lecturer at the Department of Management Information Science and Operations Management, within the Faculty of Economics and Business Administration of Ghent University. He heads the Management Information Systems research group which focuses on conceptual modeling, business ontology, business process management, and Service Science. He is also a part-time professor in software project management at the Center of Industrial Management of the Katholieke Universiteit Leuven. Dr. Poels has published in IEEE Transactions on Software Engineering, Information and Software Technology, Data & Knowledge Engineering, Journal of Systems and Software, Software and Systems Modeling, European Journal of Information Systems, Journal of Information Systems, Information Systems Journal, International Journal of Intelligent Systems, Information Sciences, Lecture Notes in Computer Science*, and* Lecture Notes in Business Information Processing.

Mario Piattini has an MSc. and PhD in Computer Science from the Technical University of Madrid and is CISA, CISM and CGEIT by ISACA (Information System Audit and Control Association), and CSQE by the ASQ. He is a professor in the Department of Computer Science at the University of Castilla-La Mancha, in Ciudad Real, Spain, where he leads the ALARCOS research group. His research interests are: Information system quality, software metrics, software maintenance and security.

APPENDIX

Classification Scheme

The following is a detailed description of the classification scheme used to analyze the extracted papers. This classification scheme was developed prior to the first round of data extraction and was subsequently refined after the pilot data was extracted and analyzed.

Type of Diagram

This dimension refers to the UML diagram that is the focus of the research in question. From UML 2.0 (OMG, 2003) onwards there are 13 different kinds of diagrams that compose UML. These 13 diagrams are a superset of the diagrams contained in previous versions of UML (referred to as UML 1.x here). The 13 types of diagrams are combined into three broad categories:

Structure diagrams emphasize the elements that must exist in the modeled system: class diagram, component diagram, object diagram, composite structure diagram (UML 2.0), deployment diagram, and package diagram.

Behavior diagrams emphasize what must happen in the modeled system: activity diagram, use case diagram, and state diagrams.

Interaction diagrams are a subset of behavior diagrams, which emphasize the control and data flow between the modeled system elements: sequence diagram, communication diagram, which is a simplified version of the collaboration diagram (UML 1.x), time diagrams (UML 2.0), and light interaction diagram (UML 2.0).

Type of Quality

There are three main model quality types: syntactic quality, semantic quality, and pragmatic quality (Lindland, Sindre, & Solvberg, 1994; Unhekkar, 2005). Each of these quality types contains some additional quality characteristics, described below. Most of these definitions are taken from (International Organization for Standardization, 1998) which is related to software product quality, or from the definitions drawn from the papers that we found during the review process related to the topic. It should be borne in mind that the definitions have been adapted to define model quality instead of software quality.

Syntactic quality refers to how well the model adheres to the rules of the language. It is also known as *syntactic correctness*. The word *correctness* refers to the absence of syntactic errors, meaning that the model is a valid instantiation of the metamodel that defines the UML type of diagram considered.

Semantic quality refers to how faithfully the modeled system is represented. There are two semantic goals: *validity* which means that all statements made in the model are correct and relevant to describe/specify the modelled system and *completeness* which means that the model contains all the statements which would be correct and relevant for describing or specifying the modelled system. There are several quality characteristics that are related to semantic quality:

- *Consistency.* The coherence between the elements of a collection. There are two types of consistency in UML: intra-model and inter-model. Intra-model consistency is all elements in a model being internally consistent and not contradicting each other. Inter-model consistency is all models in the same system being consistent with one another.

- *Completeness*. The quality that something is complete or finished. A UML model is complete when all requirements for the system being developed have been represented.
- *Correctness*. The diagram represents the system requirements adequately, without ambiguities and without redundancies in the expression.

Pragmatic quality refers to how well the model is understood. In a more general sense, pragmatics refers to the use that is made of something. The quality characteristics related to pragmatic quality are the following:

- *Maintainability*. The capability of the model to be modified. Modifications may include corrections, improvements, or adaptation of the model to changes in the system or in the system requirements. We consider this to be synonymous with the concepts of modifiability and evolvability.
- *Analyzability*. The capability of the model to be diagnosed for deficiencies or for the parts to be modified to be identified.
- *Understandability*. How well the model enables the user to understand whether the model is suitable and how it can be used for particular tasks and conditions of use. In the case of UML diagrams, understanding is related to readability, layout, and comprehension.
- *Testability*. In engineering, this refers to the capability of equipment or system to be validated. Here, whether the model can be validated to be correct.
- *Functionality.* The model's "suitability." The ability of the model to provide functions which meet stated and implied needs when the model is used under specified conditions.
- *Executability*. The executability of a model can be understood as the ability to transform a model into a software product that is executable. For example, to transform a graphical model into an executable model in XML, OWL, and so on.
- *Reusability*. The ability of some or all of an existing model to be used in the construction of a model for another system. This takes advantage of previous work, saving time and reducing redundancy.
- *Complexity*. Whether something is complex, complicated, or difficult. The complexity of a UML model is directly related to the complexity of the system that tries it tries to represent. Complex models are generally difficult to understand, reducing the ease of implementation of the final system.
- *Dependability*. It is the ability to deliver service that can justifiably be trusted. It is a term used to describe the availability performance and its influencing factors: reliability performance, maintainability performance and maintenance support performance.

Type of Evidence

In Software Engineering (SE) it is very important to increase the level of rigor and evidence in research; more importance is given to research methods that provide a scientific basis to findings (Fenton, Pfleeger, & Glass, 1994; Zelkowitz & Wallace, 1997; Zelkowitz, Wallace, & Binkley, 2003). As there are many research methods and each one provides different levels of evidence, this classification was developed using a bottom-up approach. An initial list of research methods was developed, and after reading all the papers considered in the current SLR the classification was subsequently refined. Finally, the classification was refined further by considering a similar classification described in Neto, Subramanyan, Vieira, Travassos, and Shull (2008). The different research methods are ordered by the level of evidence they support, making it possible to determine to what extent the research results are supported by empirical evidence.

Speculation. These papers describe a proposal or approach for addressing UML model quality without presenting any study or example that would indicate the feasibility of the proposal and the usefulness of the research results in practice.

Example. Consists of the description of a proposal for addressing UML model quality, where its use or application is illustrated by an example. Examples might be "toy" examples taken from books or UML models developed in real projects.

Literature review. Consists of the review of prior research to propose general frameworks, new proposals or topics for future research.

Experiment. A controlled experiment is an investigation of a testable hypothesis where one or more independent variables are manipulated to measure their effect on one or more dependent variables. Controlled experiments allow determining in precise terms how the variables are related and, specifically, whether a cause-effect relationship exists between them. Experiments are sometimes referred to as research-in-the-small, since they are concerned with a limited scope and most often are run in a laboratory setting. They are often highly controlled and hence also occasionally referred to as controlled experiments (Shull, Singer, & Sjøberg, 2008; Wohlin et al., 2000).

Case study. There is much confusion in the SE literature over what constitutes a case study. The term is often used to mean a "worked example." As an empirical method, a case study is something very different. Yin (2003) introduces the case study as "an empirical inquiry that investigates a contemporary phenomenon within its real-life context, especially when the boundaries between phenomenon and context are not clearly evident." Case studies offer in-depth understanding of how and why certain phenomena occur, and can reveal the mechanisms by which cause-effect relationships occur. Case studies are observational studies used for monitoring projects, activities and assignments. The case study is normally aimed at tracking a specific attribute or establishing relationships between different attributes. The level of control is lower in a case study than in an experiment (Shull et al., 2008; Wohlin et al., 2000).

Survey. The survey is referred to as research-in-the-large (and past) since it is possible to send a questionnaire to or interview a large number people covering whatever target population is needed. Thus, a survey is often an investigation performed in retrospect, when (for example) a tool or technique has been in use for a period of time. The primary means of gathering qualitative and quantitative data are interviews or questionnaires through a sample that is representative of the population to be studied. The results from the survey are then analyzed to derive descriptive and explanatory conclusions and are then generalized to the population from which the sample was taken (Shull et al., 2008; Wohlin et al., 2000).

The first three research methods are considered non-empirical methods, whilst the rest are considered empirical methods. Empirical methods allow researchers to determine the empirical validity of the usefulness of the research results in practice.

Type of Research Result

This dimension refers to the outcome or the "product" of the research on UML model quality. Most of the following definitions are taken from (International Organization for Standardization, 1998), which is related to software product quality, or the definitions are drawn from the papers found during the paper analysis process.

A *quality model* defines a set of characteristics, and of relationships between them, which provides a framework for specifying quality requirements and evaluating quality.

A *notation* is a system of symbolic representations of objects and ideas; it is a writing system (in fact, a formal language) used for recording concepts related with the construction of a system. Some notations that support the UML modeling can increase its expressiveness, which improves the representation of requirements and the understanding of the reader.

A *method* or algorithm is a finite sequence of instructions used to prevent or detect and delete deficiencies in models. "Method" can also be considered a technique, methodology, process, approach, or strategy.

A *tool* gives automatic support to the evaluation or assurance of quality considering different techniques (metrics, checklist, etc.).

A *metric* is a measurement scale and the method used for assessment. Metrics can be internal or external. Metrics include methods for categorizing qualitative data.

In theoretical computer science, *formal semantics* is the field concerned with the rigorous mathematical study of the meaning of programming languages and models of computation. This can be used to perform validations on models, such as consistency checking.

We consider other types of results that are not "tangible", for example a confirmation of *knowledge* (e.g., a confirmation of a theory). For example, when we replicate an experiment to confirm the findings of the original one, in reality the outcome is not "tangible," but we are confirming the knowledge acquired previously.

A *pattern* is a type of theme of recurring events or objects, sometimes referred to as elements of a set. These elements repeat in a predictable manner. It can be a template or model which can be used to generate high-quality models.

A *view* is a representation of a whole system from the perspective of a related set of concerns. The view model provides guidance and rules for structuring, classifying, and organizing architectures. Each view provides the reader with a different perspective of the system to model, which can improve the understandability of the final product.

A *checklist*, *guideline*, *rule* or *modeling convention* guides the creation of models. Such techniques attempt to obtain better models by promoting best practices, either empirically-based or scientifically proven.

Research Goal

There are several goals; in fact every researcher or organization pursues his/her/its own goals. In general, we can distinguish five different goals: understanding, measuring, evaluating, assuring, and improving the quality of UML models. These goals are:

Research into *understanding* quality seeks to define the various dimensions of quality. This research also aims at understanding the factors that impact UML model quality.

Measuring quality is concerned with developing and evaluating scales that can be used to characterise (qualitatively or quantitatively) UML model quality.

Research that *evaluates* quality investigates the relationship between quality measurements and real-world experiences with the UML model. The goal is to attach a value judgment to quality measurements.

Quality *assurance* research examines how to ensure that the process that produces the UML model actually does produce a high-quality UML model.

Finally, the research into *improving* quality examines how to increase the current quality of UML models.