Considerations about quality in model-driven engineering

Current state and challenges.

**Fáber D. Giraldo · Sergio España ·
Óscar Pastor · William J. Giraldo**

Received: date / Accepted: date

Abstract The virtue of *quality* is not itself a subject; it depends on a subject. In the software engineering field, quality means good software products that meet customer expectations, constraints, and requirements. Despite the numerous approaches, methods, descriptive models and tools, that have been developed, a level of consensus has been reached by software practitioners. However, in the model-driven engineering (MDE) field, which has emerged from software engineering paradigms, quality continues to be a great challenge since the subject is not fully defined. The *use* of models alone is not enough to manage all of the quality issues at the modelling language level.

In this work, we present the current state and some relevant considerations regarding *quality in MDE*, by identifying current categories in quality conception and by highlighting quality issues in real applications of the model-driven initiatives.

We identified sixteen categories in the definition of quality in MDE. From this identification, by applying an *adaptive sampling* approach, we discovered the five

Fáber D. Giraldo · William J. Giraldo
SINFOCI Research Group
University of Quindío, Colombia
Cra 15 Calle 12N, Armenia Quindío, Postal code: 630004.
Tel.: +57 67359300 ext 908, 995
E-mail: {fdgiraldo, wjgiraldo}@uniquindio.edu.co

Sergio España
Department of Information and Computing Sciences
Utrecht University, The Netherlands
Office: Buys-Ballotgebouw (BBL) 580 — P.O. Box 80.089 — 3508 TB
Tel: +31 616 100 939
E-mail: s.espana@uu.nl

Óscar Pastor · Fáber D. Giraldo
PROS Research Centre
Universitat Politècnica de València
Camino de Vera S/N Valencia 46022, , Spain
Tel.: +34 963877350 (Ext. 77353)
E-mail: {opastor, fdgiraldo}@pros.upv.es

most influential authors for the works that propose definitions of quality. These include (in order): the OMG standards (e.g., MDA, UML, MOF, OCL, SysML), the ISO standards for software quality models (e.g., 9126 and 25000), Krogstie, Lindland, and Moody. We also discovered families of works about quality, i.e., works that belong to the same author or topic.

Seventy-three works were found with evidence of the mismatch between the academic/research field of quality evaluation of modelling languages and actual MDE practice in industry. We demonstrate that this field does not currently solve quality issues reported in industrial scenarios. The evidence of the mismatch was grouped in eight categories, four for academic/research evidence and four for industrial reports. These categories were detected based on the scope proposed in each one of the academic/research works and from the questions and issues raised by real practitioners.

We then proposed a scenario to illustrate quality issues in a real information system project in which multiple modelling languages were used. For the evaluation of the quality of this MDE scenario, we chose one of the most cited and influential quality frameworks; it was detected from the information obtained in the identification of the categories about quality definition for MDE. We demonstrated that the selected framework falls short in addressing the quality issues. Finally, based on the findings, we derive eight challenges for quality evaluation in MDE projects that current quality initiatives do not address sufficiently.

Keywords Model-driven engineering, modelling languages, quality, quality categories, quality issues, open quality challenges.

1 Introduction

Conceptual models are the main artifacts for handling the high complexity involved in current information system (IS) development processes. The cognitive nature of the models natively supports all of the issues that are derived from the presence of several stakeholders/viewpoints, abstraction levels, and organizational challenges in an IS project. The Model-driven Engineering (MDE) is a software engineering paradigm that promotes the use of conceptual models as the primary artifacts of a complete engineering process. MDE focuses on the business and organizational concerns so that technological aspects are the result of operations over models via transformations or mappings.

An underlying foundation for working with models was proposed in the first version of the Model-driven Architecture (MDA) specification of the Object Management Group (OMG, 2003). Here the basic principles for working and managing models were defined. These can be summarized in two main features: the specification of three *abstraction levels*[1] (Computation-Independent Model - *CIM*, Platform-Independent Model - *PIM*, and Platform-Specific Model - *PSM*), and the definition of the model transformation operations. However, the increase in the number of communities of model-driven practitioners and the lack of a common consensus regarding model management (due to conceptual divergences from practitioners) has produced challenges in the usage and management of models. The MDA 1.0.1 specification has become insufficient to address these challenges

[1] Referred to *Viewpoints* in the original specification.

(see Section 2.1). Paradoxically, some of the derived challenges were formulated in IS frameworks prior to the official release of MDA specification.

One of the most critical concerns for the model-driven paradigm is the difficulty of its adoption in real contexts. Several reports have pointed out issues in model-driven adoption that are related to the misalignment between the model-driven principles and the real context (Burden et al., 2014)(Whittle et al., 2013)(Whittle et al., 2014). Some of these include the overload imposed by the model-driven tools, the lack of traceability mechanisms, and the lack of support for the adoption of model-driven strategies in organizational/development processes. Evidences from model-driven works and real applications suggest symptoms of quality assessment over models. In (Giraldo et al., 2014), the authors demonstrated the wide divergence in quality conception for MDE.

This work presents a three-year process to review the literature about the conceptualization of quality in MDE. Unlike other reviews on the same topic (most of which are summarized in (Goulão et al., 2016)), we focus on the identification of explicit definitions of quality for MDE, as well as the perception of quality in model-driven projects from real practitioners and its associated support in the academic/research field. This focus is important considering that, in the Engineering field, *high quality* is determined through an assessment that takes an artifact under evaluation and checks whether or not it is in accordance with to *its specification* (Krogstie, 2012c). Due to the specific features of the MDE paradigm, it is necessary to establish the impact of the MDE specification on the current initiatives of quality for this paradigm.

This paper presents the current state of quality conception in model-driven contexts, presenting several factors that influence it. These include the subjectivity of the practitioners, the misalignment between the real application in model-driven scenarios and the research effort required, and the implications that quality in model-driven scenarios must be considered as part of an integral quality evaluation process. This paper builds upon previous works by the authors (Giraldo et al., 2014, 2015) and makes the following contributions:

i) An analysis of the quality issues detected for both academic/research contexts and industrial contexts is performed in order to determine if current research works on quality in MDE meet the requirements of real scenarios of model-driven usage. This analysis was performed through a structured literature review using *backward snowballing* (Wohlin, 2014) on scientific publications and grey literature (non-scientific publications).

ii) A demonstration of quality in MDE issues is presented in a real scenario. This demonstration shows that current proposals of quality in MDE do not cover quality issues that are implicit in IS projects, such as the suitability in multiple-view support, the organizational adoption of modelling efforts, and the derivation of software code as a consequence of a systematic process, among others.

iii) A set of challenges that must be considered and addressed in model-driven works regarding quality and the identified categories and industrial/research alignments is presented. This set is derived from the literature reviews and should be integrally considered by any quality evaluation proposal in order to guide model-driven practitioners in how to detect and manage quality issues in MDE projects.

The remainder of this article is structured as follows: Section 2 describes quality in MDE contexts and includes an extension of a previous systematic literature review (Giraldo et al., 2014) to identify the main categories of quality conceptualization in MDE to date. Section 3 shows the results of a literature review to determine the mismatch between the quality conceptions in research and the quality conceptions of industrial practitioners and communities of model-driven practitioners. Section 4 presents a real example where multiple modelling languages are used to conceive and manage a real Information System. This real scenario highlights quality issues on modelling languages and also the insufficiency of a quality evaluation proposal in MDE for revealing quality issues in the analyzed scenario. Section 5 describes some of the challenges that quality in MDE evaluation must address based on the reported findings and evidence. Finally, Section 6 presents our conclusions.

2 Quality issues in MDE

2.1 Evolution and limitations of the MDA standard

The model-driven paradigm does not have a common conception; instead, there are a plethora of interpretations based on the goals of each model-driven community. The most neutral and accepted reference for model-driven initiative is the MDA specification which reflects the OMG vision about model-driven scenarios. It serves as a common reference for roles and operations in models.

Even though the MDA guide 1.0.1 (OMG, 2003) has been a key specification for model-driven contexts, its lack of updates over a decade has contributed to the emergence of new challenges for model-driven practitioners. Each of these challenges has been addressed by individual efforts and initiatives. Also, this guide did not provide an explicit definition about quality in models and modelling languages despite the definition of key concepts (Table 1) for using models as the main artifacts in a software/system construction process.

The MDA guide 2.0 (OMG, 2014) released in June 2014 takes into account some of the current model challenges, including issues such as *communication, automation, analytics, simulation*, and *execution*. The MDA guide 2.0 defines the implicit semantic data in the models (which is associated with *diagrams* of models) to support model management operations. Although the MDA 2.0 guide essentially preserves the basic principles of model usage and transformation, it also complements the specification of some key terms and adds new features for the management of models. Table 1 shows the differences in some of the key modelling terms between MDA 1.0 and MDA 2.0. One of the most important refinements of MDA 2.0 is the explicit definition of model as *information*.

The MDA guide 2.0 attempts to address current model challenges, including quality assessment of models through *analytics of semantic data* extracted from models (*model analytics*). However, this specification does not prescribes *how* to perform analytics of this kind or quality assessment of models.

Clearly, the refinement of key concepts that is presented in Table 1 (depicted in bold) demonstrates that the MDA guide 2.0 attempts to tackle new challenges that are implicit in modelling tasks. However, this effort is not sufficient considering that the MDA guide does not specify how to identify and manage semantic

Key term	**MDA guide 1.0.1 (2003)**	**MDA revision guide 2.0 (2014)**
System	We present the MDA concepts in terms of an existing or planned system. That system may include anything: a program, a single computer system, a combination of parts of different systems, a federation of systems (each under separate control), people, an enterprise, a federation of enterprises. Much of the discussion focuses on software within the system.	**A system is a collection of parts and relationships among these parts that can be organized to accomplish some purpose.**
Model	A model of a system is a description or specification of that system and its environment for a certain purpose. A model is often presented as a combination of drawings and text. The text can be in a modelling language or in a natural language.	**A model in the context of MDA is information that selectively** represents some aspect of a system based on a specific set of concerns. The model is related to the system by an explicit or implicit mapping. A model should include the set of information about a system to which it belongs.
Modelling language		**The structure, terms, notations, syntax, semantics, and integrity rules that are used to express a model**.
ViewPoint	A viewpoint on a system is a technique for abstraction using a selected set of architectural concepts and structuring rules in order to focus on specific concerns within that system.	**A viewpoint specifies a reusable set of criteria for the construction, selection, and presentation of a portion of the information about a system**, addressing particular stakeholder concerns.
View	A viewpoint model or view of a system is a representation of that system from the perspective of a chosen viewpoint	A view is a representation of a particular system that conforms to a viewpoint.
Abstraction		Abstraction deals with **the concepts of understanding a system in a more general way; said in more operational terms, with abstraction one eliminates certain elements from the defined scope**. This can result in introducing a higher level viewpoint at the expense of removing detail. A model is considered to be more abstract if it encompasses a broader set of systems and less abstract if it is more specific to a single system or a restricted set of systems
Platform	A platform is a set of subsystems and technologies that provide a coherent set of functionality through interfaces and specified usage patterns, which any application supported by that platform can use without concern for the details of how the functionality provided by the platform is implemented.	A platform is the **set of resources on which a system is realized**. This set of resources is used to implement or support the system.
Transformation	Model transformation is the process of converting one model to another model of the same system	**Transformation deals with producing different models, viewpoints, or artifacts from a model based on a transformation pattern**. In general, transformation can be used to produce one representation from another, or to cross levels of abstraction or architectural layers.

Table 1 Constrast of model-driven key terms between published MDA guides

data derived from models; this guide is only a preliminary (or complementary) descriptive application of model-driven standards.

In addition most of the current challenges for the model-driven paradigm have only been proposed since the emergence of previous information system frameworks by researchers. In fact, IS frameworks such as FRISCO (Falkenberg et al., 1996) (from IFIP[2]) define key aspects for the model-driven approach. These include the use of models themselves (conceptual modelling), the definition of information systems, and the use of information system denotations by representations (models), the definition of computerized information system, and the *abstraction level zero* by the presence of processors.

FRISCO gives MDA an opportunity to consider the communicative factor which is commonly reported as a key consequence of model use (Hutchinson et al., 2011b). In 1996, FRISCO suggested the need for harmonizing modelling languages and presented the *suitability* and *communicational* aspects for the modelling languages. Communication between stakeholders is critical for harmonization purposes. It allows important quality issues to be discussed from differents views (Shekhovtsov et al., 2014). FRISCO also suggested relevant features for modelling languages (*expressiveness, arbitrariness*, and *suitability*).

These kinds of FRISCO challenges produce new concerns for model-driven practitioners. For example, *suitability* requires the usage of a variety of modelling languages and *communication* requires the languages to be compatible and harmonized. Since *suitability* concludes that a diversity of modelling languages is needed, the differences between modelling languages (due to this diversity) are unjustified.

MDA was the first attempt to standarize the *model-driven* paradigm, by defining three essential abstraction levels[1] for any model-driven project and by specifying model transformations between higher/lower levels. Even though MDA has been widely accepted by software development communities and model-driven communities, the question about the ability of MDA to meet the actual MDE challenges and trends remains a pending issue.

Generally, despite the specification of the most relevant features for models and modelling languages, the lack of a specification about *when something is in MDE* is evident. This is relevant in order to be able to establish whether or not model-based proposals are aligned with the MDE paradigm beyond the presence of notational elements. There is no evidence of a quality proposal that is aligned with MDE itself.

2.2 A literature review about models and modelling language quality categories

In the RCIS 2014 conference, we first presented the preliminary results of a *Systematic Literature Review* (SR) that was performed over 21 months, with the goal of identifying the main categories in quality definition in MDE (Giraldo et al., 2014). This review is ongoing since we are attempting to demonstrate the diversity in the resulting definitions, including the most recent ones.

Fig. 1 summarizes the SR protocol that was performed, which follows the Kitchenham guidelines (Kitchenham and Charters, 2007) for ensuring a rigorous and formal search on this topic. As is depicted in Fig. 1, the protocol was enriched

[2] International Federation for Information Processing - `www.ifip.org`

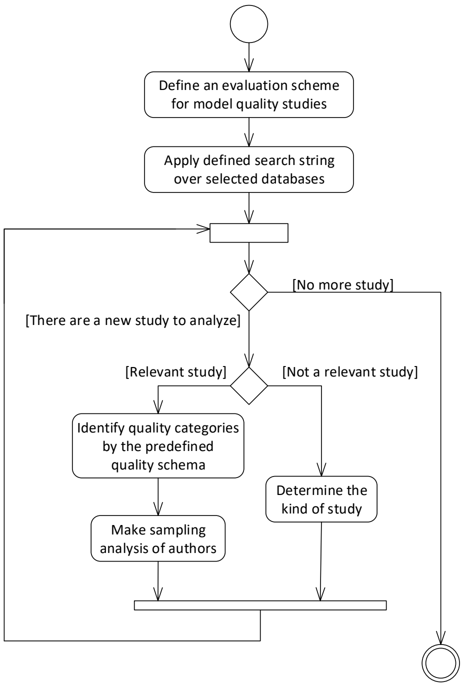

Define an evaluation scheme for model quality studies
Apply defined search string over selected databases
[No more study]
[There are a new study to analyze]
[Relevant study]
[Not a relevant study]
Identify quality categories by the predefined quality schema
Determine the kind of study
Make sampling analysis of authors

Fig. 1 Summary systematic review protocol performed.

with an *adaptive sampling* approach (Thompson and Seber, 1996) in order to find the *primary authors* on quality in MDE (see Section 2.5).

This SR addressed the following research questions:

- *RQ1*: What does *quality* mean in the context of MDE literature?
- *RQ2*: What does it mean to say that an artifact conforms to the principles of MDE?

While the main research question is RQ1, question RQ2 focuses on the fulfillment of the term *model-compliance*, i.e., whether or not the identified works have artifacts that belong to the model-driven paradigm. For this analysis, we considered modelling artifacts such as models and modelling languages. From RQ1, we derived the search string depicted as follows:

$$\begin{aligned}&Quality \wedge (Language \vee Model* \vee Modelling\ language \vee Modelling \vee Notation)\\ &\wedge (Model-driven* \vee MDD \vee MDA \vee Model-driven\ Architecture\\ &\vee\ Model-driven\ development \vee Model-based \vee MDE\ )\end{aligned}$$

The population of this work is made up of the primary studies published in journals, book sections, or conference papers, where an explicit definition about quality in model-driven contexts can be identified. The date range for this work includes contributions from 1990 until now. In order to identify these primary studies, we defined the search string that is presented above. All logical combinations were valid for identifying related works about quality in model-driven contexts. This search string was operationalized according to several configuration options (advanced mode) of each search engine. The information about the selected studies (bibliographical references) was extracted directly from the search engine.

The main sources of the studies were:

- Scientific databases and search engines such as ACM Digital Library, IEEE Explore, Springer, Science Direct, Scopus, and Willey. These include conference proceeedings and associated journals.
- Indexing services such as Google Scholar and DBLP.
- Conference Proceedings: CAISE, ER (Conceptual modelling), RCIS, ECMFA, MODELS, RE, HICSS, ECSA, and MODELSWARDS.
- Industrial repositories such as OMG and IFIP.

For this review process, a minimal set of criteria was defined in order to include/exclude studies. These are as follows:

Inclusion criteria:

- Studies from fields such as computer science, software engineering, business, and engineering.
- Studies whose title, abstract and/or keywords have at least one word belonging to each dimension of a search string (*what*, *in which*, and *where*).

Exclusion criteria:

- Studies belonging to fields that differ from computer science, software engineering, model-driven engineering, and conceptual modelling (e.g., biology, chemistry, etc.).
- Studies whose title/abstract/keywords do not have at least two dimensions of the search string configuration.
- Studies related to models in areas/fields that differ from software construction and enterprise/organizational views (e.g., water models, biological models, VHDL models, etc.).
- Studies related to artificial grammars and/or language processing.
- Studies not related to MDA/MDE/ technical spaces (Bézivin and Kurtev, 2005) (i.e., data schemas, XML processing, ontologies).

Due to the variety of studies, a classification schema was defined in order to differentiate and analyze them. Here, RQ2 plays a key role in this literature review because the evaluation of the *model-driven compliant* feature allow us to focus on the main artifacts of the modelling processes: models and modelling languages. Quality definitions are different for both artifacts. In fact, the SEQUAL framework (maybe the most complete work about quality in MDE) defines separately the *quality of models* (Krogstie, 2012c) and the *quality of modelling languages* (Krogstie, 2012b). The first definition is based on seven quality levels (*Physical, Empirical, Syntactic, Semantic and Perceived Semantic, Pragmatic, Social, and Deontic*). The second definition is based on six quality categories (*Domain appropriateness, Comprehensibility appropriateness, Participant appropriateness, Modeller appropriateness, Tool appropriateness, and Organisational appropriateness*).

Question			**Responses**
Does it offer a definition about quality?			Yes, No
If the study does	***Is the quality definition about models?***		Yes, No
	If the definition is about models	What kind of representations are referenced?	Diagram, text
	Is the quality definition about languages?		Yes, No
	If the definition is about languages	What artifacts are referenced?	Concrete syntax, abstract syntax, semantic

Table 2 Evaluation scheme applied for model quality studies in accordance with RQ2 (Giraldo et al., 2014).

All of the detected studies were analyzed using the questions in Table 2, which were defined in accordance with *RQ2*. We have resolved all of the questions that this table contains for quality studies detected. These questions identify whether or not quality studies address the scope of the *MDE compliant* feature. For studies that do not offer a quality definition, we identify the type of proposed study based on previous categories detected in our research.

2.3 Results

Table 3 presents the results of the search string applied in the databases. A second debugging process was necessary to discard studies that appear in the search results but that do not contribute to this research. This new review was made using the abstracts of the studies. These studies were considered to be *not pertinent* for this research despite their presence in the results of the search on academic databases. These works show words that are defined in the search string according to the inclusion criteria defined above; however, they do not explicitly provide any method/definition about quality in MDE and the support for multiple modelling languages. In fact, works of this kind appear as results of the search string, but they cover other topics that are aligned with model-driven approaches. We also discarded repeated studies that appear in the results of searches on multiple databases. Our analysis was made on 176 relevant studies. A summary of the analysis is presented in Fig. 2.

Source	*Total search findings*	*Useful studies*
ACM	48	7
IEEE	68	10
Springer[3]	659	59
Springer[4]	641	
ScienceDirect	166	9
Willey	371	7
Scopus [5]	784	84
	Total selected papers	**176**

Table 3 Summary of the query results in the scientific databases (October 2016).

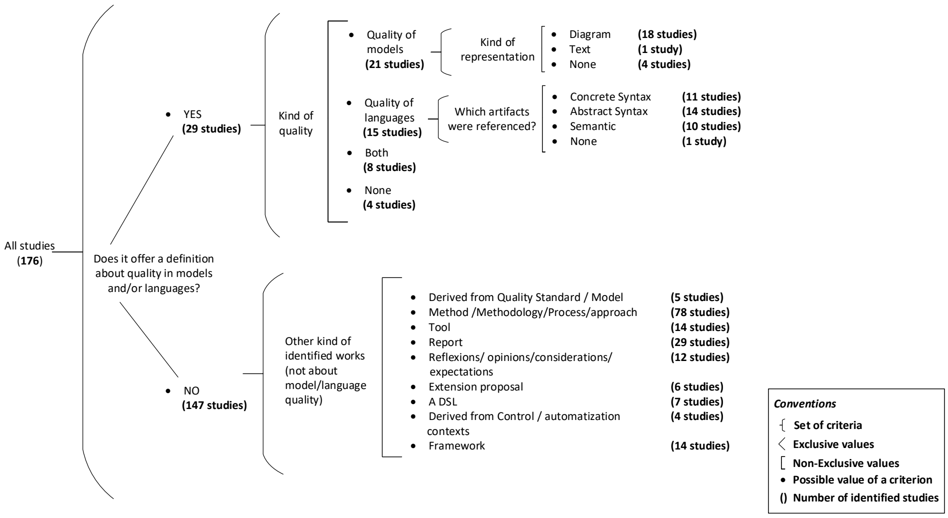

All studies
(176)
Does it offer a definition about quality in models and/or languages?
YES
(29 studies)
Kind of quality
Quality of models
(21 studies)
Kind of representation
Diagram (18 studies)
Text (1 study)
None (4 studies)
Quality of languages
(15 studies)
Which artifacts were referenced?
Concrete Syntax (11 studies)
Abstract Syntax (14 studies)
Semantic (10 studies)
None (1 study)
Both
(8 studies)
None
(4 studies)
NO
(147 studies)
Other kind of identified works (not about model/language quality)
Derived from Quality Standard / Model (5 studies)
Method /Methodology/Process/approach (78 studies)
Tool (14 studies)
Report (29 studies)
Reflexions/ opinions/considerations/ expectations (12 studies)
Extension proposal (6 studies)
A DSL (7 studies)
Derived from Control / automatization contexts (4 studies)
Framework (14 studies)
Conventions
{ Set of criteria
< Exclusive values
[Non-Exclusive values
• Possible value of a criterion
() Number of identified studies

Fig. 2 Summary of identified studies that offer definition about quality in modelling languages, in response to RQ2.

This debugging is particularly important because it reflects the broad implications involved in the terms *model* and *quality*. Although these discarded works are model-driven compliance, they reflect the ambiguity that *model-driven compliance* represents (even without full MDA compliance), so the mere existence of models may be criteria enough to determine compliance with the model-driven paradigm. Also, the generality in the use of the terms *model* and *quality* in the software engineering context and related areas is demonstrated, producing a diversity of works to support initiatives under those terms as a result.

During the analysis of the 176 primary studies reviewed, we checked whether each paper offered an explicit definition of quality, or at least if the study provided a conceptual framework that would allow a definition of quality to be derived as a result of the application of some theory. Therefore, from the 176 detected studies, we detected 29 studies (16.48% of the target population) that provide a definition of quality in model-driven contexts. The number of papers that provide a definition of quality is relatively low with respect to the number of identified and debugged studies. This indicates that the quality concept leads to works where quality is the result of the application of a specific approach. In those cases quality is reduced to specific dimensions (e.g., metrics, detection of defects, increased productivity, cognitive effectiveness, etc.).

Of the 29 studies that provide definitions about quality, 21 studies (11.93% of all studies) offer a definition in terms of quality of models. Eighteen of these studies (10.23%) present the quality of models in terms of diagrams (mostly UML), and only one study (0.57%) defines the quality of textual models. In addition, 15

[3] *with Computer Science discipline / Engineering subdiscipline*

[4] *with Business Management discipline / SWE subdiscipline*

[5] *with Computer Science category*

of the 29 quality studies (8.52%) offer a definition of quality at the modelling language level, of which 11 studies (6.25%) mention quality at the concrete syntax level, 14 studies (7.95%) at the abstract syntax level, and 10 studies (5.68%) at the language semantics level. Of the 29 quality studies, 8 studies (4.55%) were detected in which the quality definition is shared between models and modelling languages. Similarly, we detected 4 other studies (2.27%) whose definitions of quality do not consider model or language artifacts. These studies are associated to *Category 1* presented in Section 2.4, which proposes a quality model for a quality framework for a specific model-driven approach.

On the other hand, 147 studies were detected (83.52% of total identified studies) that do not provide an explicit definition of quality in model-driven contexts. The presence of these studies is a consequence of specific model-driven proposals formulated to promote specific works on specific aspects of quality such as methodological frameworks, experiments, processes, etc. Of these works:

- Five studies (2.84%) present specific adoptions of standards such as ISO 9126, ISO 25010, descriptive models such as CMMI©, and approaches such as Goal-Question-Metric (GQM) to support the operationalization of techniques applied in model-driven contexts (including model transformations).
- Seventy-eight of the 176 identified studies (44.32%) have proposed methodologies to perform tasks in model-driven contexts that are commonly framed in quality assurance processes (e.g., behavioral verification of models, performance models, guidelines for quality improvement in the transformation of models, OCL verifications, checklists, model metrics and measurement, etc).
- Fourteen studies (7.95%) report tools that are built to evaluate and/or support the applicability of specific quality initiatives in model-driven contexts.
- Twenty-nine studies (16.48%) are about designed experiments or empirical procedures to evaluate quality features of models that are mostly oriented towards their understandability.
- Twelve studies (6.82%) reported specific dissertations about quality procedures in model-driven contexts such as data quality, complexity, application of agile methodology principles, evaluation of languages, etc.
- Six studies (3.41%) are works that extend predefined model-driven proposals such as metamodels, insertion of constraints into the complex system design processes, definition of contracts for model substitutability, model-driven architecture extension, etc.
- Seven studies (3.98%) propose domain-specific languages (DSL) for specific tasks that are related to model management or model transformations.
- Four studies (2.27%) report model-driven experiences in industrial automation contexts where models become useful mechanisms to generate software with a higher level of quality which is defined as the presence of specific considerations at the modelling level previous to the software production.
- Fourteen studies (7.95%) define frameworks for multiple purposes such as measuring processes, quality of services, enrichment of languages, validation of software implementations according to their design, etc.

The existence of these studies indicate that the terms *quality* and *model* are often used as pivots to highlight specific initiatives that cover only certain dimensions of quality and MDE.

ID Study	Year	Type of study	Study	Valida-ted?	Operatio-nalized	Quality definition
1	***2011***	BC	(Escalona et al., 2011)	NO	NO	*Category 1* Quality model for MDWE
2	***2011***	BC	(Espinilla et al., 2011)	YES	YES	
3	***2010***	CP	(Domínguez-Mayo et al., 2010)	NO	YES	
4	***2011***	JA	(Domínguez-Mayo et al., 2011)	NO	NO	
5	***2012***	BC	(Krogstie, 2012c)	NO	YES	*Category 2* SEQUAL framework
6	***2012***	BC	(Krogstie, 2012d)	NO	YES	
7	***1995***	CP	(Krogstie et al., 1995)	NO	YES	
8	***2009***	JA	(Mohagheghi et al., 2009a)	NO	NO	*Category 3* 6C framework
9	***2008***	M	(Mohagheghi and Dehlen, 2008a)	NO	NO	
10	***2007***	WP	(Mohagheghi and Aagedal, 2007)	NO	NO	
11	***2009***	BC	(Hindawi et al., 2009)	YES	NO	*Category 4* UML guidelines
12	***2007***	BC	(Lange, 2007b)	NO	YES	*Category 5* Model size metrics
13	***2010***	CP	(Amstel, 2010)	NO	NO	*Category 6* Quality in model transformations
14	***2005***	T	(Merilinna, 2005)	YES	YES	
15	***2011***	JA	(Grobshtein and Dori, 2011)	YES	NO	
16	***2012***	JA	(Chaudron et al., 2012)	NO	NO	*Category 7* Empirical evidence about the effectiveness of modelling with UML
17	***2005***	WP	(Lange and Chaudron, 2005)	YES	YES	
18	***2010***	JA	(Cruz-Lemus et al., 2010)	NO	NO	*Category 8* Understandability of UML
19	***2008***	JA	(Heymans et al., 2008)	NO	YES	*Category 9* Application of model quality frameworks
20	***2003***	M	(Atkinson et al., 2003)	NO	NO	*Category 10* Quality from structural design properties
21	***2013***	WP	(Mijatov et al., 2013)	NO	YES	
22	***2014***	WP	(López-Fernández et al., 2014)	YES	YES	*Category 11* **(new)** Quality of metamodels
23	***2013***	WP	(Le Pallec and Dupuy-Chessa, 2013)	YES	YES	
24	***2007***	JA	(Morais and da Silva, 2015)	NO	NO	*Category 12* **(new)** Formal quality methods
25	***2014***	JA	(Heidari and Loucopoulos, 2014)	NO	NO	*Category 13* **(new)** Quality factors of business process models
26	***2015***	BC	(Reijers et al., 2015)	NO	NO	
27	***2015***	CP	(Maes and Poels, 2007)	YES	NO	*Category 14* **(new)** Quality procedures derived from IS success evaluation framework
28	***2012***	CP	(Sayeb et al., 2012)	NO	YES	*Category 15* **(new)** A quality patterns catalog for modelling languages and models
29	***2015***	JA	(Challenger et al., 2015)	YES	NO	*Category 16* **(new)** An evaluation framework for DSMLs that are used in a specific context

Table 4 Summary of identified studies about quality in MDE (updated to October 2016).

2.4 Identified categories of the definition of *quality* in MDE

In this research, a *category* is a set of established practices, activities, or procedures for evaluating the quality of models, regardless of any formality level and the modelling languages involved. According to RQ1, a summary of the defined categories is presented in Table 4[6]. The categories reflect the grouping of the quality works identified. In contrast to the previous report of (Giraldo et al., 2014), in this extension we found six new categories for quality in MDE, which are highlighted in Table 4.

- **Category 1 - Quality model for MDWE**: This quality model defines and describes a set of quality criteria (usability, functionality, maintainability and reliability) for the model-driven web approach (MDWE). The model also defines the weights for each element of the quality criteria set, and the relation of the elements with the user information needs (MDE, web modelling, tool support and maturity).
- **Category 2 - SEQUAL framework**: This is a semiotic framework that is derived from the initial framework proposed by Linland et al. Quality is discussed on seven levels: *physical, empirical, syntactic, semantic, pragmatic, social*, and *deontic*. The way different quality types build upon each other is also explained.
- **Category 3 - 6C framework**: These works propose the 6C quality framework, which defines six classes of model quality goals: *correctness, completeness, consistency, comprehensibility, confinement, and changeability.* This framework emerges as a grouping element that contains model quality definition and modelling concepts from previous works such as Lindland, Krogstie, Sølvberg, Nelson. and Monarchi.
- **Category 4 - UML guidelines**: In this work the quality of a model is defined in terms of style guide rules. The quality of a model is not subject to conformance to individual rules, but rather to statistical knowledge that is embodied as threshold values for attributes and characteristics. These thresholds come from quality objectives that are set according to the specific needs of applications. From the quality point of view, only deviations from these values will lead to corrections; otherwise, the model is considered to have the expected quality. While the style guide notifies the user of all rule violations, non-quality is detected only when the combination of a set of metrics reach critical thresholds.
- **Category 5 - model size metrics**: Quality is defined in terms of model size metrics (MoSMe). The quality evaluation considers defect density through model size measurement. The size is generally captured by the height, width, and depth dimensions. This already indicates that one single size measure is not sufficient to describe an entity.
- **Category 6 - quality in model transformations**: The work presented in (Amstel, 2010) defines the quality of model transformation through internal and external qualities. The internal quality of a model transformation is the quality of the transformation artifact itself. The quality attributes that describe

[6] The following conventions are used in the *Type of study* column of Table 4 : BC [*Book Chapter*], CP [*Conference Proceeding*], JA [*Journal Article*], WP [*Workshop Proceeding*], T [*Thesis*], M [*Monograph*].

the internal quality of a model transformation are: understandability, modifiability, reusability, modularity, completeness, consistency, and correctness. The external quality of a model transformation is the quality change induced on a model by the model transformation. The work proposes a direct quality assessment for internal quality and an indirect quality assessment approach for external quality, but only if it is possible to make a comparison between the source and the target models.
Other work that is associated to this category is presented in (Merilinna, 2005). This work proposes a specific tool that automates the quality-driven model transformation approach proposed in (Matinlassi, 2005). To do this, the authors propose a procedure that consists of the development of a rule description language, the selection of the most suitable CASE tool for making the transformations, and the design and implementation of a tool extension for the CASE tool.
In addition, in the work presented in (Grobshtein and Dori, 2011) quality is a consequence of an OPM2SysML view generation process, that uses an algorithm with its respective software application. Thus, quality is defined as the effectiveness and fulfillment of faithfully translating OPM to SysML.

- **Category 7 - empirical evidence about the effectiveness of modelling with UML**: The identified works do not provide a definition for quality in models; it contains a synthesis of empirical evidence about the effectiveness of modelling with UML, defining it as a combination of positive (benefits) and negative (costs) effects on overall project productivity and quality. The work contributes to the quality in models by showing the need for quality assurance methods based on the level of quality required in different parts of the system, and including consistency and completeness dimmensions as part of quality assurance practices as a consequence of the communicational purposes of (UML) models.
- **Category 8 - understandability of UML**: This is an empirical study that evaluates the effect that structural complexity has on the understandability of the UML statechart diagram. The report presents three dimensions of structural complexity that affect understandability. The authors also define a set of nine metrics for measuring the UML statechart diagram structural complexity. This work is part of broad empirical research about quality in modelling with UML diagrams where works like (Piattini et al., 2011) can be identified.
- **Category 9 - application of model quality frameworks**: This is an empirical study that evaluates and compares feature diagrams lenguages and their semantics. This method relies on formally defined criteria and terminology based on the highest standards in engineering formal languages defined by Harel and Rumpe, and a global language quality framework: the Krosgtie'SEQUAL framework.
- **Category 10 - quality from structural design properties**: Quality assurance is the measurement of structural design properties such as coupling or complexity based on a UML-oriented representation of components. The UML design modelling is a key technology in MDA, and UML design models naturally lend themselves to design measurement. The internal quality attributes of relevance in model-driven development are structural properties of UML artifacts. The specific structural properties of interest are coupling, complexity, and size. An example is reported in (Mijatov et al., 2013) where the authors

propose an approach to validate the *functional correctness* of UML activities by the executability of a subset of UML provided by the fUML standard.

- **Category 11 - quality of metamodels**: Works of this kind specific languages and tools to check desired properties on metamodels and to visualize the problematic elements (i.e., the non-conforming parts of metamodels). The validation is performed over real metamodel repositories. When the evaluation is done, feedback is delivered to both MDE practitioners and metamodel tool builders.
- **Category 12 - formal quality methods**: This category is related to the ARENA formal method reported in (Morais and da Silva, 2015) that allows the quality and effectiveness of modelling languages to be evaluated. The reported selection process was performed over a set of user-interface modelling languages. The framework is a mathematical formula whose parameters are predefined properties that are specified by the authors.
- **Category 13 - quality factors of business process models**: In (Heidari and Loucopoulos, 2014) the authors proposed the QEF (Quality Evaluation Framework) method to assess the quality of business processes through their models. This method could be applicable to any business process notation; however, its first application was reported in BPMN models. The framework relates and measures business process quality factors (like resource efficiency, performance, reliability, and etc) that are the inherent property of a business process concept and can be measured by quality metrics.
  In this category, the SIQ framework (Reijers et al., 2015) is also identified for the evaluation of business process models. Here, three categories for evaluating models are distinguished: syntactic, semantic, and pragmatic. By this, there is an inevitable association of SIQ with previous quality frameworks such as SEQUAL (*Category 2*) and some works of Moody; however, the authors clarify that the SIQ categories are not the same as those that were previously defined in the other quality frameworks. The authors show how SIQ is a practical framework for performing quality evaluation that has links with previous quality frameworks. SIQ attempts to integrate concepts and guidelines that belong to the research in the BPM domain.
  A complete list of works around quality for business process modelling is presented in (de Oca et al., 2015). This works reports a systematic review for identifying relevant works that address quality aspects of business process models. The classification of these works was performed by the use of the CMQF framework (Nelson et al., 2012), which a combination of SEQUAL and the Bunge-Wand-Weber ontology.
- **Category 14 - quality procedures derived from IS success evaluation framework**: The authors in (Maes and Poels, 2007) proposed a method to measure the quality of modelling artifacts through the application of a previous framework of Seddon (Seddon, 1997) for evaluating the success of information systems. The method proposes a selection of four related evaluation model variables: Perceived Semantic Quality (PSQ), Perceived Ease Of Understanding (PEOU), Perceived Usefulness (PU) and User Satisfaction (US). This method is directly associated with a manifestation of the *perceived semantic quality* (*Category 2*) described in (Krogstie et al., 1995).
- **Category 15 - a quality patterns catalog for modelling languages and models**: The authors in (Sayeb et al., 2012) propose a collaborative pattern

system that capitalizes on the knowledge about the quality of modelling languages and models. To support this, the authors introduce a web management tool for describing and sharing the collaborative quality pattern catalog.

- **Category 16 - an evaluation framework for DSMLs that are used in a specific context**: the authors in (Challenger et al., 2015) formulate a specific quality evaluation framework for languages employed in the context of multi-agent systems (MAS). Their systematic evaluation procedure is a comparison of a modelling proposal with a hierarchical structure of dimension / sub-dimension / criteria items. The lower level (criteria) defines specific MAS characteristics. For this category, *quality* is a dimension that has two sub-dimensions: the general DSML assessment sub-dimension (with criteria such as domain scope, suitability, domain expertise, domain expressiveness, effective underlying generation, abstraction-viewpoint orientation, understandability, maintainability, modularity, reusability, well-written, and readability) and the *user perspective* sub-dimension (with criteria such as developer ease, and advantages/disadvantages). Both sub-dimensions are addressed by qualitative analysis; it is assumed that this type of analysis is performed with case studies that are designed with experimental protocols.

2.5 Adaptive sampling

Using the principles of the *adaptive sampling* approach defined in (Thompson and Seber, 1996), we analyzed the identified papers in order to explore clustered populations of studies about quality in models. We made a review of the bibliographical references of each study detecting *reference authors* or *works* (i.e., previous studies formulated before the publication of the analyzed study that have been cited in the quality studies identified). We established the *reference authors* or *reference works* as those who have been referenced by at least two quality studies detected of different authors.

To do this, we defined Tables 5 and 6, where the rows refer to the authors of the identified quality studies and the columns contain the referenced authors or works. A link in the *(i,j)* cell on Tables 5 and 6 (the color black in the cell fill) indicates that the author of the *j* column has influenced the authors of the *i* row; so that the *i*-author(s) cite the *j*-author(s) in the quality study(ies) that were analyzed.

In the Table 5 the columns (or *j*-authors) correspond to the same authors of quality studies; this was intentionally done in order to show the influence of authors on the analyzed quality studies. Table 5 shows that Krogstie (*Category 2*) is the author that has had the most influence on the quality works analyzed. His work influences 50% of the identified quality studies, followed by Lange (*Category 5*) with 31.3%. Two special cases occur in the columns of Krogstie and Mohagheghi (*Category 3*); they appear as authors of identified quality papers, but they were cited by other works that were not detected in the searches of the academic databases. We wanted to highlight the other works of the authors that influence the analyzed studies.

Table 5 also shows the studies that are referenced, created, or influenced by works of the same author. These studies do not affect other authors or proposals for quality in models. However, Table 5 also shows quality communities of researchers

	Authors of identified studies																			
Studies Authors ⇒ Studies Authors ⇓	*Cat. 1*	*Cat. 2*	*Cat. 3*	*Cat. 4*	*Cat. 5*	*Cat. 6 Study 13*	*Cat. 6 Study 14*	*Cat. 6 Study 15*	*Cat. 7*	*Cat. 8*	*Cat. 9*	*Cat. 10 Study 20*	*Cat. 10 Study 21*	*Cat. 11*	*Cat. 12*	*Cat. 13 Study 25*	*Cat. 13 Study 26*	*Cat. 14*	*Cat. 15*	*Cat. 16*
Category 1	■		■							■										
Category 2		■														■	■			
Category 3		■	■		■							■								
Category 4				■	■															
Category 5					■				■											
Category 6 Study 13		■	■		■	■														
Category 6 Study 14							■													
Category 6 Study 15								■												
Category 7					■				■	■										
Category 8										■		■			■					
Category 9		■									■									
Category 10 Study 20												■								
Category 10 Study 21													■							
Category 11														■						
Category 12		■													■					
Category 13 Study 25																■	■			
Category 13 Study 26		■															■			
Category 14		■																■		
Category 15		■							■										■	
Category 16			■																	■

Table 5 Sampling of categories' authors in quality frameworks for MDE (Part I).

	Reference authors or works					
Reference Authors ⇒ Studies Authors ⇓	(Lindland et al., 1994)	(Moody et al., 2002; Moody and Shanks, 2003; Moody, 2005, 2006)	OMG	ISO/IEC (2001)	(Kitchenham et al., 2002; Kitchenham and Charters, 2007)	(Harel and Rumpe, 2000, 2004)
Category 1				■		
Category 2	■	■	■			
Category 3	■		■	■	■	■
Category 4				■		■
Category 5			■		■	
Category 6 Study 13	■			■		
Category 6 Study 14			■			
Category 6 Study 15			■			
Category 7						
Category 8			■		■	
Category 9	■	■	■			■
Category 10 Study 20			■	■		
Category 10 Study 21			■			
Category 11						
Category 12	■	■				
Category 13 Study 25			■			
Category 13 Study 26	■	■				
Category 14			■	■		
Category 15	■	■		■		
Category 16				■	■	

Table 6 Sampling of categories' authors and reference authors or works in quality frameworks for MDE (Part II).

on topics such as model metrics and guidelines mainly applied over UML. Works led by Lange, Chaudron and Hindawi contribute to the consolidation of these research communities. This community phenomenon was originally reported in (Budgen et al., 2011), and is described in works like (Lange and Chaudron, 2006; Lange et al., 2003; Lange and Chaudron, 2005; Lange et al., 2006). In fact, the works of Lange presented in (Budgen et al., 2011) suggest that most model quality problems are related to the design process, which shows that a conflict arises with all viewpoint-based modelling forms, and not just UML.

In the Table 6 the columns represent other authors or works which were identified in the review of the bibliographical references for each quality study. As Table 6 shows, the OMG specifications and ISO 9126 standard are the most important industrial references that influence the formulation of quality studies.

The OMG specifications were cited by 68.8% of the authors of identified categories. The OMG specifications that were most cited by authors were MDA[7] specification followed by UML, MOF, OCL, and SysML specifications. Evidence of the adoption of the OMG standard suggests that the works are MDA compliant, but this does not necessarily means an explicit adoption or alignment to the MDA initiative itself. The ISO standards (cited by 50% of the works) are used to support quality model proposals on the taxonomy composed by features, sub-features, and

[7] http://www.omg.org/mda/

quality attributes. It is even useful for evaluation purposes. This kind of adoption excludes the quality dimensions that are involved in the ISO standards (quality of the process, intern quality, extern quality, and quality in use).

Linland's quality framework (Lindland et al., 1994) is one of the reference frameworks that is most frequently used and cited by the authors of the primary studies (43.75%). This framework was one of the first quality proposals formulated and it takes into account the syntactic, semantic, and pragmatic qualities regarding goals, means, activities, and modelling properties. The Krosgtie quality framework (an evolution from Linland's framework) is recognized as the work with the most influence on contemporary works about the quality of models. In the case of Krogstie and Moody (cited by the 31.25% of the works), the authors of the analyzed studies cited early papers where they began to present the first versions and applications of their approaches. Finally, it is important to highlight the references to Kitchenham's works to support the application of systematic review guidelines and analysis in procedures on empirical software engineering.

2.6 Other findings

As a consequence of the searches performed, an identification of studies belonging to the same authors or topics was made. These were sets of related works with specific approaches for evaluating quality over models such as model metrics, defect detections, cognitive evaluation procedures, checklists, and other works about quality frameworks. For our research, this distinction is particularly important because of their presence in the search results; however, most of them do not contribute a formal definition for quality in models. Instead, they focus on specific topics that are considered in quality strategies.

The identified families are the following:

- Understandability of UML diagrams (Piattini et al).
- SMF approach (Piattini et al).
- NDT (University of Sevilla Spain)
- SEQUAL Framework (Krogstie)
- Constraint - Model verification (Cabot et al, and others) (Chenouard et al., 2008; González et al., 2012; Tairas and Cabot, 2013; Planas et al., 2016).
- fUML (Laurent et al., 2013)(Mayerhofer, 2012).
- OOmCFP (Pastor et al) (Marín et al., 2010)(Marín et al., 2013)(Panach et al., 2015a).
- 6C Framework (Mohagheghi et al).

These families show how the interpretation of quality is reducted to specific proceedings or approaches in a way similar to mismatches or limitations on the term software quality. Because of this, some authors like (Piattini et al., 2011) suggest the need for more empirical research to develop (at least) a theoretical understanding of concept of quality in model.

2.7 Discussion

Section 2.4 answered *RQ1* (the meaning of quality in the MDE literature). The obtained categories of quality were classified in accordance with the schema that was defined in Table 2 (derived from *RQ2*). Despite the many model-driven works, tools, modelling languages, etc., the concept of quality has only been ambiguously defined by the MDE community. Most quality proposals are focused primarily on the evaluation of UML for many varied interests and goals.

Works about quality in MDE are limited to specific initiatives of the researchers without having applicability beyond the research or specific works considered. This contrasts with the relative maturity level of quality definitions such as the one presented in Section 2.2 (SEQUAL framework).

The low number of works on quality and the diversity of quality categories reflect specific quality frameworks and the respective communities that support these quality concept. The high number of results in the searches performed indicates misconceptions about quality due to the wide spectrum of model enginering in terms of its ease of application (any model can conform to MDE), and the lack of mechanisms to indicate when something is in accordance with MDE.

There are many definitions on quality in models in the literature, but, there is also dispersion and a general disagreement about quality in MDE contexts; this is demonstrated by multiple categories in the quality in MDE presented in Section 2.4.

MDE requires a definition of quality that is aligned with the principles and main motivations of this approach. Extrapolation of software quality approaches alone are insufficient because we move from a concrete level (code production, software quality assurance activities) to a higher abstract level to support specific modelling domains.

Traditional evaluations of UML are not enough for a full understanding of quality in models; UML is oriented to functional software features and also, is an object-oriented modelling approach. UML is the defacto software modelling approach, but the evaluation of quality models in terms of UML excludes the overall spectrum of MDE initiatives. Quality evaluation of cognitive effectiveness could restrict the overall quality in models to the diagram and notational levels.

The quality proposals analyzed do not consider how to reduce the complexity added by the model quality activities (experiments, changes in syntax and semantics, evaluation of quality features of a high level of abstraction, etc).

The quality evaluation categories reported do not take into account the implications at the tool level. Tools are a particularly important issue because a language can be explained by its associated tool. New challenges related to the tools that support MDE initiatives have emerged; an example can be seen in (Köhnlein, 2013). In the proposals, tools are limited to validation cases without further applicability beyond the proposal itself. Also, the lack of reports about the validation and use of the quality proposals demonstrates the level that they were formulated in preliminarly stage of research.

2.8 The relationship between quality in MDE and V&V

Verification and validation procedures (commonly referred to as V&V) are key strategies in the software quality area for avoiding, detecting, and fixing defects and quality issues in software products. These procedures are applied throughout all the lifecycle of the software product before its release.

MDE also takes advantage of V&V procedures by applying them in modelling artifacts (i.e., languages, models, and transformations) in order to find issues before the generation of artifacts such as source code or other models. One of the most representative examples in the MDE literature of V&V procedures is the MoDEVVa[8] (Model Driven Engineering, Verification and Validation) workshop of the ACM/IEEE MODELS conference.

Thirteen of the sixteen categories of quality in MDE are associated to specific V&V procedures in MDE reported by the authors, highlighting the studies reported in (Mijatov et al., 2013) - *Category 10* - and (López-Fernández et al., 2014) - *Category 11* - which appear in the proceedings of the MoDEVVa workshop (MoDEVVa 2013 and MoDEVVa 2014, respectively). Three categories (*2*, *3*, and *15*) provide guidance for evaluating quality in modelling artifacts. Works of these categories must be interpreted in order to be applied in specific evaluation scenarios.

3 A mismatch analysis between industry and academy field on MDE quality evaluation

Quality in models and modelling languages has been considered in several ontological IS frameworks even before the formulation of the model-driven architecture (MDA) specification by the Object Management Group (OMG), as mentioned above. The ISO 42010 standard (612, 2011) defines that the architecture descriptions are supported by models[9], but it recognizes that the *evaluation of the quality* of the architecture (and its descriptions) is the subject of further standarization efforts.

The survey artifact proposed in the CMA workshop of the MODELS conference[10] presents a set of key features for all modelling approaches, considering issues related to the modelling paradigm involved, the notation, views, etc. This is a valuable effort to harmonize the study of the modern modelling approaches, which suggest higher features to analyze in modelling languages. However, some key issues such as usability, expressiveness, completeness, and abstraction management (which are key in ontological frameworks) are poorly described. The support for transformations between models, the role of tools in a model-driven context, and the diagrams as main interaction mechanism between models and users also require better descriptions..

The above evidence demonstrates quality in MDE is not an unknown factor for the adoption of model-driven initiatives in real contexts, e.g., software, IS, or complex engineering development processes. Therefore, the consideration and/or use of the MDE paradigm in industrial scenarios is an important source for detecting quality issues, taking into account that it would impact the adoption of model-

[8] Current version of the MoDEVVa workshop available in `https://sites.google.com/site/modevva/`. Previous versions can be accessed in `https://sites.google.com/site/modevva/previous-editions`.

[9] For ISO 42010, the *architecture* of a system is the *essence* or *fundamentals* of it expressed through models.

[10] `http://www.cs.colostate.edu/remodd/v1/sites/default/files/ComparisonCriteria-v3.pdf`.

driven initiatives. It is also important to identify the support of the current MDE quality proposals for the model-driven industrial communities and practitioners.

For this reason, we performed a complementary literature review in order to find evidence of the *mismatch* between the research field of modelling language quality evaluation and actual MDE practice in industry. In (Giraldo et al., 2015), we presented the preliminary results of a literature review. This search is currently ongoing.

3.1 Literature review process design

We have performed a structured literature review using the *backward snowballing* approach. It has been demonstrated that it yields similar results to search-string-based searches in terms of conclusions and patterns found (Jalali and Wohlin, 2012), and we did not want to miss valuable grey literature in the results. Grey literature is not published commercially and is seldom peer-reviewed (e.g., reports, theses, technical and commercial documentation, scientific or practitioner blog posts, official documents), but it may contain facts that complement those of conventional scientific publications.

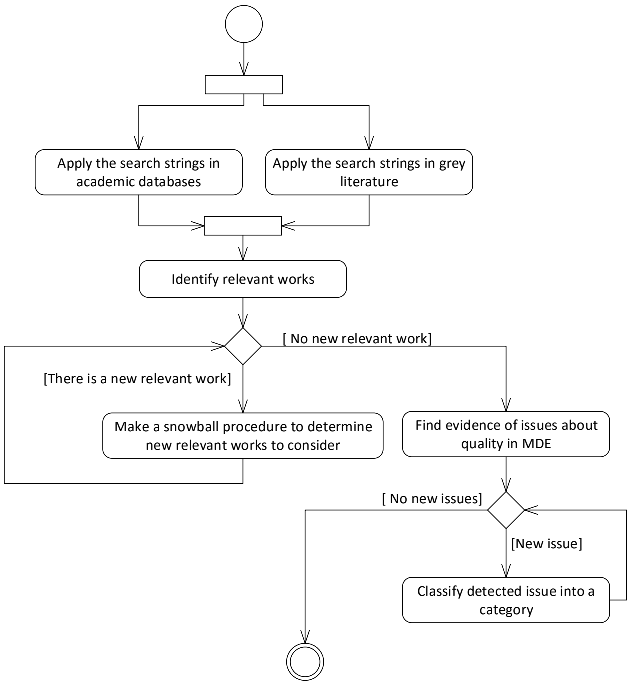

Apply the search strings in academic databases
Apply the search strings in grey literature
Identify relevant works
[No new relevant work]
[There is a new relevant work]
Make a snowball procedure to determine new relevant works to consider
Find evidence of issues about quality in MDE
[No new issues]
[New issue]
Classify detected issue into a category

Fig. 3 Summary of the literature review protocol performed.

Fig. 3 summarizes the literature review protocol that was performed. This literature review is an extension of a previous systematic review reported in Section 2.2. The *snowballing* sampling approach helps to identify additional works from an initial reference list. This list was obtained from an initial keyword search. We use the snowballing procedure reported in (Wohlin, 2014) to address the following research questions:

- *RQ1*: What are the main issues reported in MDE adoption for industrial practice that affect modelling quality evaluation?
- *RQ2*: What is the focus of works on modelling quality evaluation in the corresponding research field?
- *RQ3*: Does the term *model quality evaluation* have a similar meaning in both the industrial level and the academic/research level?
- *RQ4*: Is there a clear correspondence between industrial issues of modelling quality and trends in the identified research?

Our snowballing search method was performed as follows:

1. The initial searches were done on scientific databases and search engines such as Scopus, ACM Digital Library, IEEE Explore, Springer, Science Direct, and Willey[11]. These include conference proceedings and associated journals. We used search strings depicted as follows:

$$(MDE \vee Model{-}driven*) \wedge (real\ adoption \vee adoption\ issues \vee problem\ report\ )$$

2. For the resulting works, we chose articles that show explicit reports about the applicability of the MDE paradigm in real contexts.
3. For those relevant works, quality issues were identified, and their reference lists were reviewed to find related works on reporting quality issues. This iteration was made until no new works were identified.
4. To complement the quality issues detected, we analyzed web portals of software development communities, such as blogs, technical web sites, forums, social networks, and portals accessed from Google web search, using similar strings regarding previous scientific database searches. Our goal was to identify model quality manifestations from software practitioners who work with specific technical and business constraints.

Several inclusion/exclusion criteria were applied on the search results to identify relevant works for our analysis. These criteria are as follows:

Inclusion criteria

- Works where an explicit manifestation of quality on a model-driven issue were included and presented. Examples of these manifestations are *model transformation tool problems, misalignment of model-driven principles with specific business concerns, skepticism of the model-driven real application and sufficiency*, among others.
- Reports that include an approach to identify model-driven issues in real applications (e.g., interviews with people that perform roles within an IS project, questionnaires, or description about real experiences).

[11] Currently, search engines such as Scopus could reference other main databases, but we preferred to check the above-mentioned databases to avoid the loss of valuable reports.

- Works that relate (and/or perform) a literature review approach on the applicability of model-driven approaches in real scenarios.
- For non-academic works (web portals), we checked the impact and quality of the posted information. This was done by reviewing the forum messages, the academic references used, and the level of the community that supports those portals in terms of technological reports, conference-related mentions, and participants' profiles.
- For non-academic works (web portals), we checked the link between authors and participants with well-known companies that report the application of model-driven approaches (e.g., MetaCase, Mendix, Integranova and etc.), and academic/industrial conferences related to model-driven and IS topics (e.g., CodeGeneration Conference, RCIS, CAiSE, MODELS, and etc.).

Exclusion criteria

- Works that report application cases of model-driven compliance approaches or initiatives (notations, application on a specific domain, guidelines, etc.), but whose main focus is the promotion of those specific approaches, without considering the collateral effects of their application.

Each included work was analyzed in order to find quality evidence (i.e., explicit sentences) in the adoption of the model-driven approach reported. Because of the kind of works detected and the level of formality of their sources, it was necessary to access the full content of each work, in order to determine the relevance of each contribution regarding the expectations formulated in our research questions. Despite the common terms used in the search strings, we only accepted works based on the MDE applicability report.

More information about reported quality issues can be found in the technical report available in (Giraldo et al., 2016). This report presents all the works with their associated statements that support the detected quality issues. During the review of these issues, we found that quality evidence could be categorized as follows:

Industrial issues (RQ1)

- Industrial issue 01: *Implicit questions derived from the MDE adoption itself.*
- Industrial issue 02: *Organizational support for the MDE adoption.*
- Industrial issue 03: *MDA not enough.*
- Industrial issue 04: *Tools as a way to increase complexity.*

Academic/research issues (RQ2)

- A/R issue 01: *UML as the main language to apply metrics over models and defect prevention strategies.*
- A/R issue 02: *Hard operationalization of model-quality frameworks.*
- A/R issue 03: *Software quality principles extrapolated at modelling levels.*
- A/R issue 04: *Specificity in the scenarios for quality in models.*

Sections 3.2 and 3.3 describe in depth the above categories related to *RQ1* and *RQ2*, respectively. Section 3.4 presents the results of the mismatch related to *RQ3* and *RQ4*.

3.2 Detected categories for industrial quality issues

In response to *RQ1*, in the following, we present four categories that we defined for grouping the sentences of industrial quality issues. In (Giraldo et al., 2016), 240 quality sentences are reported from industrial sources. These affect the perception of model-driven initiatives, and, therefore, their quality. Each category groups sentences of several sources that share a common quality issue. These categories were used to facilitate the analysis of the industry-academy mismatch.

The *MDA is not enough* category groups the sentences that report the lack of the MDA specification to resolve questions in the use and application of models and modelling languages (see Section 2.1). The *Implicit questions derived from the MDE adoption itself* category groups sentences in which open questions remain unresolved when a model-driven initiative (with its associated set of languages, models, transformations, and tools) is applied in a specific context.

The *Tools as a way to increase complexity* category groups the sentences that report explicit problems in the use and application of model-driven tools (e.g., tools based on the Eclipse EMF-GMF frameworks and associated projects). Tools are the main mechanism for creating and managing models by the application of modelling languages. Finally, the *Organizational support for the MDE adoption* category groups the sentences that report issues in the organizational adoption of model-driven initiatives.

In the following, we describe each category in more detail:

3.2.1 MDA is not enough

As a reference architecture, MDA provides the foundation for the usage and transformation of models in order to generate software using three predefined abstraction levels. A definition of quality in models that is supported in the alignment with MDA would not be enough. This is because the compliance with the guidelines of this architecture is the minimum criterion expected for the management of models and it must be implicitly supported by current tools and model-driven standards.

A real consequence of this MDA insufficiency is presented in (Hutchinson et al., 2014). The authors show the lack of consensus about the best language and tool as being a pending issue that is not covered in the MDA specification. This issue affects real scenarios where a combination of languages is used to support specific industrial tasks. The model-driven community have recognized the lack of structural updates of the MDA specification in the last decade, which produces imprecise semantic definitions over models and transformations (Cabot). The MDA revision guide 2.0 (OMG, 2014) released in June 2014 preserves these issues.

3.2.2 Implicit questions derived from the MDE adoption itself

This covers concerns about the suitability of languages and tools (Hutchinson et al., 2014)(Staron, 2006), new development processes derived from MDE adoption (Hutchinson et al., 2014), MDE deployment (Hutchinson et al., 2011a), the scope of the MDE application (Aranda et al., 2012; Whittle et al., 2014), and implicit questions about how and when a MDE approach is applied, e.g., *when and where to apply MDE ?* (Burden et al., 2014), and *which MDE features mesh* *most easily with features of organizational change? which create most problems?* (Hutchinson et al., 2011a). The correct usage of the modelling foundation in current modelling approaches is also questioned (Whittle et al., 2014).

3.2.3 Tools as a way to increase complexity

The absence of support for MDE tools and the lack of trained people require that great effort be made to adapt to the context of the organization with probably less that optimun results (Burden et al., 2014). This issue leads to problems with the followings: customization, tailoring, and interoperability among modelling tools (Burden et al., 2014)(Mohagheghi et al., 2013b), management of traceability with several tools (Mohagheghi et al., 2013b), the high level of expertise and effort required to develop a MDE tool (Burden et al., 2014)(Mohagheghi et al., 2013b), tool integration (Baker et al., 2005)(Burden et al., 2014)(Mohagheghi and Dehlen, 2008b)(Mohagheghi et al., 2013a), the dissatisfaction of MDE practicioners with the available tools (Tomassetti et al., 2012), the lack of technological maturity of the tools (Mohagheghi et al., 2013a), the scaling of the tools to large system development (Mohagheghi and Dehlen, 2008b), poor user experience (Mohagheghi et al., 2009b), too many dependencies for adopting MDE tools (Whittle et al., 2013), and poor performance (Baker et al., 2005).

3.2.4 Organizational support for the adoption of MDE

This category represents issues that are related to commitments, costs especially training (Hutchinson et al., 2014), resistant to change (Aranda et al., 2012), the alignment and adaptation of MDE with *how* people and organizations work (Burden et al., 2014)(Whittle et al., 2014), and organizational decisions based on diverging expert opinions (Hutchinson et al., 2011b).

The main concern of these works is the misalignment between the model-driven principles and the organizational elements. Most of the works on model-driven compliance are related to technical adoption, such as modelling tools, model-transformation consistency, and the incorporation of models in software development scenarios. However, due to the lack of an explicit model-driven process, organizational issues may not be able to be completely managed in a model-driven approach, by final model users.

3.3 Detected categories for academic/research quality issues

In response to *RQ2*, we propose another four categories in order to group the focus of the works on quality evaluation in the academic/research field. Seventy-one issues from this field were reported in (Giraldo et al., 2016). The categories reflect the intention of the researchers in the model-driven field for managing quality issues. These are as follows:

3.3.1 Hard operationalization of model-quality frameworks

High abstraction and specific model issues influence the operationalization of model quality frameworks (i.e., the instrumentation of a framework by a software tool). Therefore, quality rules or procedures may not be fully implemented by operational mechanisms such as XSD schemas, EMF Query support, etc. In (Störrle and Fish, 2013) present an attempt to make operational the *Physics of notations* evaluation framework (Moody, 2009), however this operationalization (and any similar proposal) could be ambiguous as a consequence of the lack of precision and detail of the framework itself.

An example of model quality assurance tools as reported in (Arendt and Taentzer, 2013) where an operational process for assessing quality through static model analysis is presented. Instead of having an operational model quality framework, a quality framework like *6C* (Mohagheghi et al., 2009a) has been used as a conceptual basis for deriving a quality assurance tool.

The lack of full operationalizations of model quality evaluation frameworks shows that model evaluation is still more an art than science (Nelson et al., 2005), and that current specifications to evaluate quality in models and modelling languages continue to be complex procedures for language designers and final model users.

3.3.2 Defects and metrics mainly in UML

Most of the quality proposals in models focus their effort on the applicability of metrics in UML models and the definition of guidelines to detect and avoid defects in UML diagrams. This trend is a direct consequence of the limitation of the model-driven paradigm in UML terms.

Limitations are based on the specific model-driven vision of OMG. This promotes the model-driven approach in UML, which offers a set of modelling notations that cover multiple aspects of business and systems modelling. MDA also promotes the UML extension using profiles by tailoring the core UML capabilities in a unified tooling environment (OMG, 2003, 2014).

However, this vision contrasts with the low incidence of UML as the main artifact in software and IS development processes. Clear and recent evidence is reported in (Petre, 2013), where the main trend regarding the use of UML among a group of software experts was *No Usage (No UML)*; the second representative trend was *UML models were useful artifacts for specific and personal tasks, but these were discarded after explanatory tasks were completed.* A very low number of participant experts mention UML in code-generation tasks. This vision also contrasts with recent evidence of removing UML in recognized development environments due to its lack of use as reported in (Krill, 2016).

Ambiguity in UML persists due to the specific meanings and interpretations that model practitioners applied to it. This ambiguity directly affects the full adoption of UML as a standard for software and information systems development communities. Also, there is no link between the quality issues reported in UML with the standardization effort of UML by OMG. The complexity in the UML formal specifications contributes to the confusion of model-driven practitioners.

3.3.3 Specificity in the scenarios for quality in models

The most relevant works in this issue have a specific focus from which the quality of models are defined. The quality frameworks formulated in (Krogstie, 2012a; Lindland et al., 1994) have a semiotic foundation due to the use of signs in the process of the domain representation. Other works like (Mohagheghi and Dehlen, 2008a; Mohagheghi et al., 2009a) propose desirable features (goals) for models. Some proposals are specific to the scope of the research performed (e.g. (Domínguez-Mayo et al., 2010)).

Some of the classical procedures for verifying the quality of conceptual models are related to the cognitive effectiveness of notations (generally UML models). In this way, quality motivations are limited to an evaluation (and probably intervention) process on a notation.

3.3.4 Software quality principles extrapolated at modelling levels

Within the MDE literature there are proposals that extrapolate specific approaches for evaluating software quality at model levels, which are supported by the fact that *MDE is a focus of software engineering.* Some of the reported software quality approaches include the usage of metrics, defect detection in models, application of software quality hierarchies (in terms of characteristics, sub-characteristics and quality attributes), best practices for implementing high quality models and model transformations. There is even a research area that is oriented to the evaluation of the usability of modelling languages (Schalles, 2013), where the usability in diagrams is prioritized as the main quality attribute of models.

The main motivation for this extrapolation is the level of *relative maturity* of the software quality initiatives. In (Moody, 2005) the author suggests the formulation of quality frameworks for conceptual models based on the explicit adoption of the ISO 9126 standard, because of its wide usage in real scenarios and the fact that this standard makes recognizable the properties of a product or service. In (Kahraman and Bilgen, 2013) authors present a set of artifacts that are formulated to support the evaluation of domain-specific languages (DSLs). These instruments are derived from an integration of the CMMI model, the ISO 25010, standard and the DESMET approach. The success of a DSL is defined as a combination of related characteristics that must be collectively possessed (by combining practices from CMMI and ISO 25010 hierarchy). Proposals of this kind assume that there is an existing relation among organizational process improvement efforts, their maturity levels, and the quality of DSL's.

Software quality involves a strategy for the production of software that ensures user satisfaction, absence of defects, compliance with budget and time constraints, and the application of standards and best practices for software development. However, software quality is a ubiquitous concern in software engineering (Abran et al., 2013), and therefore, in the MDE context, additional effort is required for the adoption of the MDE approach.

3.4 Findings in the literature review of mismatch

For this literature review journal papers were the main source of quality issues for both contexts (industrial and research), as shown in Fig. 4. However, for the *industrial* context, specialized websites (grey literature[12]) make significant contributions to the quality from a practitioners' perspective. We found 49 industrial

[12] Grey literature refers to documents that are not published commercially and that are seldom peer-reviewed (e.g., reports, theses, technical and commercial documentation, scientific

	MDE Quality evidence in industry	MDE Quality evidence in academic & research
Industrial issue 01	(Vara and Marcos, 2012; Cuadrado et al., 2014; Hutchinson et al., 2014; Bertrand Portier, 2009; Haan, 2008; Whittle et al., 2014; Dubray, 2011; Linders, 2015; Tone, 2010; Finnie, 2015; Klinke, 2008; Vallecillo, 2014; Hebig and Bendraou, 2014; Brown, 2009; DenHaan, 2009, 2011b; Pierson, 2007; Cabot, 2009; Brambilla, 2016; OMG, 2016; Mohagheghi et al., 2009b; Igarza et al., 2012; Clark and Muller, 2012; Whittle et al., 2015; Aranda et al., 2012; Baker et al., 2005; Whittle et al., 2013; Cuadrado et al., 2014; Hoang, 2012; Platania, 2016; Fournier, 2008)	(Agner et al., 2013; Panach et al., 2015a; Davies et al., 2006; Mussbacher et al., 2014; Quintero and Muñoz, 2011; Quintero et al., 2012; Poruban et al., 2014; Bruel et al., 2015; Picek and Strahonja, 2007)
Industrial issue 02	(Brambilla and Fraternali, 2014; Lukman et al., 2013; Hutchinson et al., 2014; Torchiano et al., 2013; Mohagheghi et al., 2013b; Bertrand Portier, 2009; Whittle et al., 2014; Dubray, 2011; Klinke, 2008; Kulkarni et al., 2010; Vallecillo, 2014; Igarza et al., 2012; Aranda et al., 2012; Baker et al., 2005; Quora, 2014)	
Industrial issue 03	(Brambilla and Fraternali, 2014; Lukman et al., 2013; Bertrand Portier, 2009; Haan, 2008; Dubray, 2011; Corneliussen, 2008; Klinke, 2008; Kulkarni et al., 2010; Vallecillo, 2014; Brown, 2009; Mora et al., 2006; Ortiz et al., 2013; Clark and Muller, 2012; Hutchinson et al., 2014; Platania, 2016; Krill, 2016; Quora, 2015a,b)	(Mussbacher et al., 2014; Singh and Sood, 2009)
Industrial issue 04	(Cachero et al., 2007; Torchiano et al., 2013; Haan, 2008; Linders, 2015; Corneliussen, 2008; Klinke, 2008; Kulkarni et al., 2010; Vallecillo, 2014; DenHaan, 2009, 2011b; Pierson, 2007; Brambilla, 2016; Igarza et al., 2012; Ortiz et al., 2013; Whittle et al., 2015; Hutchinson et al., 2014)(DenHaan, 2010, 2011a; Marín et al., 2014; Aranda et al., 2012; Baker et al., 2005; Whittle et al., 2013; Cuadrado et al., 2014; Hoang, 2012; Quora, 2015b)	(Teppola et al., 2009; Mussbacher et al., 2014; Quintero and Muñoz, 2011; Quintero et al., 2012; Poruban et al., 2014; Bruel et al., 2015; Singh and Sood, 2009)
A/R issue 01		(Agner et al., 2013; Davies et al., 2015; Nugroho, 2009; Gorschek et al., 2014; Laguna and Marqués, 2010; Wehrmeister et al., 2014; Dijkman et al., 2008; Picek and Strahonja, 2007)
A/R issue 02		(Fabra et al., 2012; Teppola et al., 2009; Molina and Toval, 2009; Nugroho, 2009; Kessentini et al., 2013; Dijkman et al., 2008; Vallecillo, 2010)
A/R issue 03		(Panach et al., 2015b; Molina and Toval, 2009; Kolovos et al., 2008)
A/R issue 04		(Panach et al., 2015b; Fabra et al., 2012; Van Der Straeten et al., 2009; Vallecillo, 2010)

Table 7 The works found in the review of the mismatch between the research field of modelling language quality evaluation and the actual MDE practice in industry. Last update: October 2016.

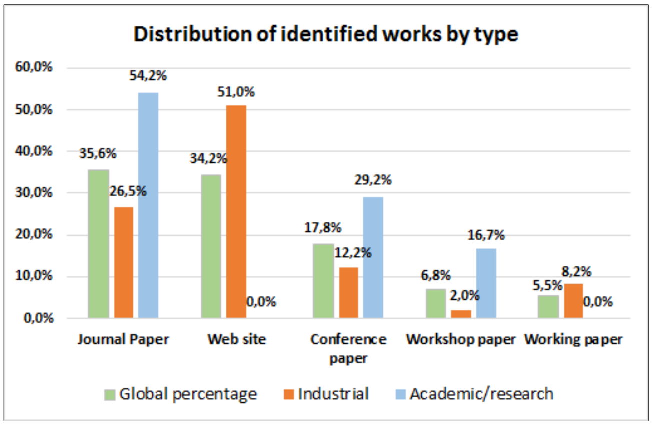


Fig. 4 Percentage distribution of identified works by type.

works and 24 academic/research works; the analysis was made on a total of 73 works.

To answer *RQ3*, Table 7 presents the identified works classified in the categories described in Section 3.1. The found mismatches show that model-driven practicioners perceive quality of models and modelling languages in different ways. It greatly depends on the application context where modelling approaches are used.

Fig. 5 shows the percentage of quality issues detected in the industrial works analyzed. From a real software engineering perspective, there is an initial assumption about the high degree of impact related to model-driven tools and its consequences on development and organizational environments. However, for the industrial works analyzed, we detected the *implicit questions derived from the MDE adoption itself* issue as being the first concern of quality regarding the applicability of models and modelling languages. This issue is derived from the great ambiguity about when something is in MDE (or when something is MDE compliant) and also from the open questions generated in the application of models.

Clearly, industrial publications show a marked trend when discussing the deficiency, consequences, and support of the modelling act itself before using of specific modelling tools. In addition, quality issues related to the tools are evident in the detected works. Beyond the consequences of the application of model-driven initiatives, tools become a key artefact in perceiving, measuring and managing quality issues in modelling languages, taking into account concerns related to organizational, interactional, and technical levels.

The results in Fig. 6 highlight the presence of academic and research works that address industrial issues such as *implicit questions derived from the MDE adoption itself* and *tools as a way to increase complexity*. Some statements from academic and research sources show an alignment with industrial issues. However, in Fig. 6, the percentage of works that address industrial issues is lower than the sum of the percentages of works that promote specific interests of researchers in this field.

or practitioner blog posts, official documents). It may contain facts that complement those of conventional scientific publications.

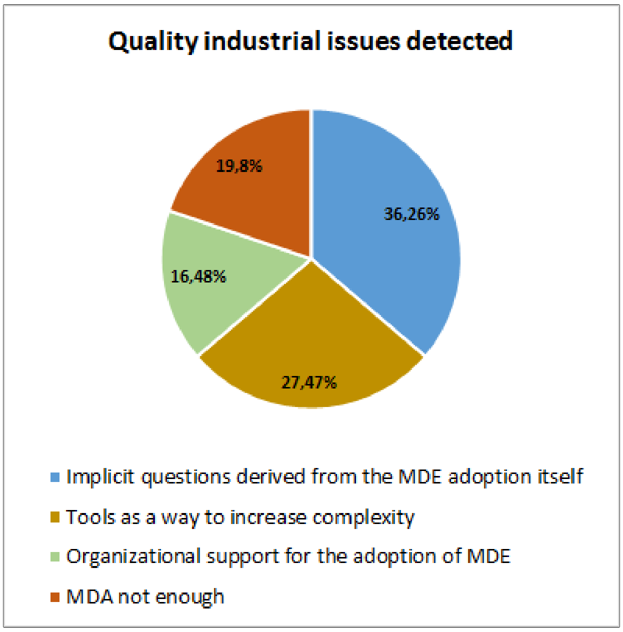


Fig. 5 Percentage of quality industrial issues detected.

It shows that model-driven researchers tend to focus on theoretical works; thus, these industrial issues are not interesting or relevant to model-driven researchers. This lack of research support increases the conceptual and methodological gaps for the real application of model-driven initiatives and promotes confusion in the model-driven paradigm.

An example of this theoretical emphasis of researchers is the relative proximity of the issue of *implicit questions derived from the MDE adoption itself* of the industrial category and the issue of *defects and metrics mainly on UML* of the academic/research category. There are many efforts that target the quality management of models through the intervention of modelling practices in UML as the defacto language for software analysis/design. There is clearly a gap between these quality trends and the reports about the real usage and applicability of UML, as in the study reported in (Petre, 2013).

Academic/research works also consider the inherent complexity involved in achieving concrete tools from theoretical quality frameworks for models and languages due to the high level of abstraction involved in them. In contrast, industrial works do not report specific quality issues that are related to the academic/research categories. Therefore, for answering *RQ4*, the above evidence demonstrates a very significant difference between the perceptions and efforts regarding quality in modelling languages and models for industrial and academic/research scenarios. This issue gap between industrial and academic communities requires a method that resolves the problems in industry that are not covered by the current methods.

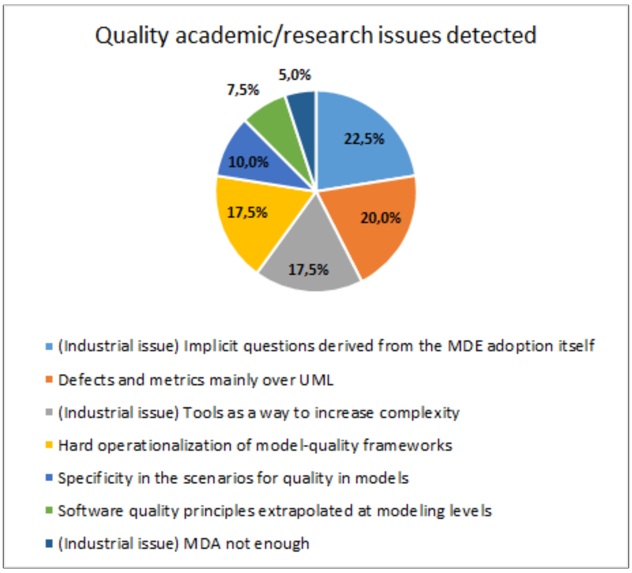


Fig. 6 Percentage of quality academic/research issues detected.

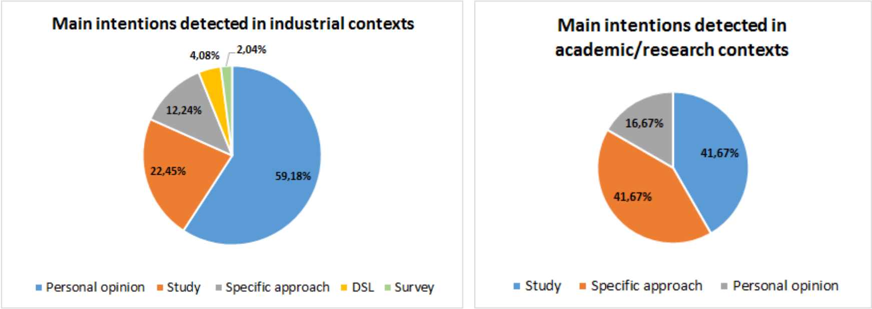


Fig. 7 Summary of the intentions found in the analyzed works.

In the academic-research and industrial contexts, the subjectivity and the particularities of the application scenarios play an important role in the derivation of quality issues in model-driven initiatives. Fig. 7 shows the main intention of the analyzed works, depending on whether the work was written for academic/research purposes or for industrial purposes. These intentions refer to personal opinions, studies, or approaches. The main sources for the industrial context are *opinions* and *interactions in web sites* reported in the grey literature. This is valuable considering that these resources show real experiences of attempts to use model-driven initiatives in real software projects.

In the academic and research field, there is a strong trend (41.67% of reported works) towards specific model-driven initiatives promoted by practitioners.

Among these initiatives are DSL, model-driven approaches, operations on models (e.g., searching over models, establishing the level of detail of models), and specific considerations for model transformations (e.g., BPMN models to petri nets). Although several modelling language quality issues were extracted from formal studies performed by researchers, it is important to note how quality issues also serve as *excuses* (or *pivots*) for promoting specific model-driven initiatives.

In summary, the current academic/research methods have not solved quality issues for MDE reported in the industry (Section 3.2). It seems that researchers have not yet addressed these problems satisfactorily[13]. Therefore, we consider it necessary to list the open challenges and to define (in a greater depth) the research roadmap proposed in (Giraldo et al., 2015) in order to cover these issues comprehensively. Thus, in Section 4 we show a real scenario in which quality issues associated to the Sections 3.2.2 and 3.2.4 are depicted. Afterwards, in the Section 5 we present a set of challenges we inferred from the evidences related in both above literature reviews.

4 The sufficiency of current quality evaluation proposals

In this section, we present a scenario for multiple application modelling languages. The case presented in this section was a finished project that had been previously developed by the authors, the implementation of an information system for institutional academic quality management. In this IS project, quality issues were empirically demonstrated. Quality evaluation methods were not used during the execution of this model-driven project. The full specification of the case is presented in Appendix A.

The objective of this scenario is to demonstrate that the application of an existing quality method has not revealed all of the modelling quality issues of the project, despite the execution of the analysis as a post-mortem task. For this empirical study, we have chosen the *Physics* of Notations - PoN - (Moody, 2009), the most widely cited modelling language quality evaluation framework available in the literature. We show that, despite having many useful features, this framework is insufficient to cover all the needs that arise when evaluating the quality of (sets of) modelling languages in MDE projects. The identification of these uncovered needs serves as additional input for the definition of a research roadmap in Section 5.

A *post-mortem* analysis was performed to evaluate the quality of a set of modelling languages that were employed in the project (Flowchart, UML, E/R, and architecture languages). Section A.1 presents the models that were obtained in the project. Each one of the PoN principles was applied to the obtained models in the project to determine whether or not the models meet the PoN principles. Section A.2 presents the results of the quality assessment with the PoN framework.

Table 8 summarizes the detected quality issues in the proposed scenario. Although it is true that the application of the PoN framework allows quality issues in the modelling scenario to be detected, other critical quality issues were not detected by this method. PoN meets its goals of analyzing the concrete syntax of the modelling languages under evaluation. However, other quality issues appear for

[13] Some attempts and efforts have been made such as (Mussbacher et al., 2014), but quality issues continue to be open challenges.

Id	Description of the quality issue	Detected?	PoN principle
I1	All business process activities are included in the same diagram. This hinders model comprehensibility, especially in the case of domain experts.	Yes	Complexity management.
I2	For the modelling language that was used for modelling the business processes (Flowchart), 81.25% of its symbols were not used.	Yes	Semiotic clarity - Symbol Deficit.
I3	The original semantics of some modelling elements was altered by the domain experts.	Yes	Semiotic clarity - Symbol Excess.
I4	The interpretation of business processes models was affected by the low number of symbols used for modelling them.	Yes	Perceptual Discriminability.
I5	Some modelling elements have a meaning that is not generally correct.	Yes	Semantic Transparency semantically opaque.
I6	A lack of guidance was detected to denote the process flow of the diagrams.	Yes	Visual Expressiveness.
I7	Excessive textual representation in the business diagrams.	Yes	Dual coding
I8	The graphical notation used for business process models is too minimalist.	Yes	Graphic Economy
I9	The Flowchart language is not suitable for modelling business processes or complex systems because of its simplicity.	No	
I10	Final users of the IS suggest the use of BPMN to model the institutional academic quality management process.	No	
I11	The analysts decided to use a UML profile for business process modelling when there are modelling languages that are better suited for this purpose. In fact, in later stages of the project, domain experts happened to discover BPMN and contested the initial language selection.	No	
I12	A decoupling between models from organizational and system levels was found.	No	
I13	There are no mechanisms to manage traceability of modelling artifacts.	No	
I14	Extra effort was required to translate non-UML system models to specific platforms.	No	

Table 8 Quality issues detected for the multiple modelling language scenario.

factors such as multiple modelling languages, different abstraction levels, several stakeholders, and viewpoints.

One single quality framework may be insufficient to integrally address all quality issues in MDE projects. Even though there are guidelines to support the application of existing individual quality methods which avoid subjective criteria that influence the final results of the analysis for PON (e.g.,(da Silva Teixeira et al., 2016)), there are no systematic guidelines for using quality methods for MDE in combination.

5 Open challenges in the evaluation of the quality of modelling languages in MDE contexts

Sections 2, 3, and 4 presented the problems and questions that remain regarding the evaluation of quality issues in the MDE field. Current phenomena for model-driven applicability, use, and the associated quality issues create several challenges that impact the adoption of the model-driven paradigm. Here it is not enough to evaluate quality from a prescriptive perspective as is proposed for most of the identified quality categories of Section 2.4. Any quality evaluation method in models and modelling languages requires the incorporation of the realities regarding MDE itself.

These realities are not unfamiliar to the model-driven community. In the following, we have highlighted the terms and sentences that represent them in bold. They were taken from recognized sources that provide definitions about models. A quick overview of some classical model definitions reveals the presence of *subject* as a fundamental element of the model itself. This is valid for the unified axiom of model as concept in order to understand a subject or phenomenon in the form of description, specification, or theory:

- OMG MDA guide 1.0 (OMG, 2003): A model of a system is **a description or specification of that system** and its **environment** for a certain **purpose**. A model is often presented as a combination of drawings and text. The text may be in a **modelling language** or in a natural language. Model is also a **formal specification** of the **function, structure and/or behavior** of an application or system.
- OMGA MDA guide 2.0 (OMG, 2014): **A model is information** that selectively represents an aspect of a system based on a specific **set of concerns**. A model should include the **set of information** about a system that is within.
- ISO 42010-2011 (612, 2011): A model can be anything: a model can be a **concept** (a mental model), or a model can be a **work product**. Every model has a **subject**, so **the model must answer questions about this subject**.
- Stanford Encyclopedia of Philosophy (SEP) (Hodges, 2013): A model is a construction of a **formal theory** that describes and explains a **phenomenon**. You model a system or structure that you plan to build by writing a **description** of it.

A conceptual foundation for the model-driven approach was established for the information system community before the formulation of MDA itself, taking into account the main challenges (see Section 2.1). ISO 42010 established the importance of the *viewpoint, view, model kind,* and *architectural description* concepts. Also, the term *correspondence* must be used in the specification of model transformations. Specifically, FRISCO presents the *suitability* and *communicational* aspects for the modelling languages and the need for harmonization of modelling languages. The communicative factor is commonly reported as a key consequence of model usage (Hutchinson et al., 2011b).

In addition, the *subject* of modelling includes quality issues as presented in Section 3. The subjective usage of model representations, the freedom to formulate model-driven compliance initiatives, and the wide applicability of models for any IS-supported domain, requires an underlying support to analyze models and all the artifacts that modelling languages provide in order to model any IS phenomena. This rationale must consider the key premises on which the model-driven context was promoted. These become the main input for any model analytics process, in a way that is complementary to previous model quality evaluation frameworks.

The research roadmap of (France and Rumpe, 2007) was (and continues to be) widely accepted by model-driven practitioners due to their explicit skepticism about the MDE vision and its related problems (including quality evaluation issues). Other roadmaps as presented in (Kolovos et al., 2013; Mohagheghi et al., 2009b; Rios et al., 2006; Vallecillo, 2010) address specific concerns about MDE applicability, with informal considerations about its the adoption in real scenarios, and lack of relation to any IS foundations. These quality issues that we have described in Section 3 show a gap between the real application of MDE and its foundational principles.

Because of the divergence of the quality definition in MDE, the lack of support from the academic/research field for practitioners of the model-driven initiatives, and the diverse interpretations by the different research communities, we have deduced a set of challenges that any quality evaluation method for MDE should consider in order to assess quality from a MDE viewpoint (i.e., taking into account the main realities that govern this paradigm). We consider that a required rationale for quality evaluation in model-driven initiatives must address the following critical challenges in the MDE paradigm itself:

5.1 Using multiple modelling languages in combination

This reality is inherent to IS development where multiple views must be used to manage the concerns derived from stakeholders. Each view could have its associated language, and in the same way, one language could support several views of the information system. In this case, if L is the set of all the modelling languages $\{l_1, l_2, \ldots l_n\}$ used to support the views (and viewpoints) in a IS project and Q is the assessment of quality for MDE, then $Q\{L\} \neq Q\{l_1\} \cup Q\{l_2\} \cdots \cup Q\{l_n\}$.

Several questions are derived from IS feature: the *suitability* of the languages used to model and manage a specific view, the *coverage* level of the modelling proposals, the relevance and pertinence regarding the specific intention of modelling, and the degree of utility of a modelling language by virtue of the stakeholder concerns under consideration.

Even though, the evaluation of these features heavily depends on subjective criteria, their consideration is mandatory to be able to support modelling and integration approaches on views within a model-driven project (with their respective implications). Subjectivity is intrinsic to the model-driven paradigm, and although an absolute truth in model-driven will not be possible, its consideration facilitates the consolidation of model management strategies in model-driven environments. These quality questions are the essential for information systems.

The treatment of the *multi*-factor concept is not a new topic in the MDE community. It has been considered in previous MDE challenges as reported in

(Van Der Straeten et al., 2009). However, the percentage of works that propose a method to manage the *multi* modelling phenomenon is very low (Giraldo et al., 2014) and these do not provide a computerized (operational) tool for model-driven practioners.

The *multiple* feature in models and information systems (and its derived quality implications) inherently leads to the analysis of the capabilities provided by modelling languages to represent an IS phenomenon adequately and to integrate it with other proposals that cover others IS concerns. The current information systems foundations provide the required inference tools to contrast the capabilities of modelling languages to support the *multiple* feature.

In Section 2.3, the percentage of identified works that consider quality evaluation methods for multiple languages is low (4.02%). It shows the minor impact of quality in model proposals on the management of complex information system developments, which contain multiple views and viewpoints supported by conceptual models.

The works that consider evaluation over a set of modelling languages (Krogstie, 2012c,d; Mohagheghi et al., 2009a) present two theorical evaluation frameworks whose operationalizations are not clear (i.e., any evaluation procedure could be too abstract for the MDE community especially for people from software development contexts). However, their works are a very important advance in the foundation of a body of knowledge for quality in MDE. The evaluation of multiple modelling languages remains an open issue. Evidence can be found in different reports of the application of some quality works. Generally, these reports present the evaluation of a single modelling language. The evaluation of multiple languages is empirically deducted.

5.2 Assessing the compliance of modelling languages with MDE principles

There is a general consensus about the MDE concept as the promotion of models as primary artifacts for software engineering activities (Di Ruscio et al., 2013)(González and Cabot, 2014), and as the presence of model transformations that refine abstract/concrete modelling levels. However, due to the generality of this consensus, an initiative may be model-driven without a strict fulfillment of the minimum aspects necessary for real applicability with technological support (e.g., notations without an associated abstract syntax, stereotyped elements of common modelling languages, or modelling proposals with specific intentions and poor adoption by model-driven practitioners).

Despite the specification of the most relevant features for models and modelling languages, there is a lack of specification about *when something is in MDE* (Section 3.2.2); this must be established if model-based proposals are aligned with the MDE paradigm beyond the presence of notational or textual elements. There is no quality proposal that is aligned with MDE itself (i.e., a quality approach that defines a validation procedure to determine whether or not a model-driven initiative meets the MDE core features). Although intuitively one could consider that it all boils down to the extent to which a specific model-driven method meets the core MDE features, literature on quality has not explicitly covered in detail what it means to be aligned with MDE and whether the quality of this alignment can be measured.

It is arguable that the existence of methods claiming to be model-driven that do not actually fulfill the MDE paradigm influences the stakeholders perception of the MDE paradigm itself. For instance, en alleged method might not fulfill expectations, and these negative experiences might end up being generalized to the paradigm itself. This can be a factor that hinders the adoption of MDE approaches and contributes to open issues such as the ones covered in Section 3.2.

The definition about *when something is in MDE* or *when something is MDE compliant* must take into account critical concerns beyond the simple usage of models or textual and graphical representations. This includes the alignment of the model with a modelling purpose (in a way similar to the multidimensional views in IS development), the explicit association with an abstraction level (principle that is introduced by MDA), the conceptual support of modelling languages through metamodels, and the capabilities provided by the modelling artifact to integrate with other modelling initiatives and to support models transformations, mappings, and software generation.

In this way, quality (Q) can be defined as the operation $Q = \{L, E\}$, where L is a set $\{l_1, l_2 \ldots l_n\}$ of one or more modelling languages in a MDE project and E represents a MDE environment (i.e., the set of the concerns in a MDE project such as the one described above). Therefore, determining Q implies that $\forall\ l \in L,\ l$ satisfactorily meets (or addresses) E.

5.3 Explicitly using abstraction levels as quality filters of modelling languages

This challenge is a consequence of the MDA specification where three abstraction levels (Computation-Independent Model (CIM), Platform-Independent Model (PIM), and Platform-Specific Model (PSM)) were explicitly proposed in order to clarify and define the usage and scope of models with regard to their intention and closeness with business, system or technical levels.

Abstraction levels act as the reference element to evaluate the convenience of modelling proposals. Harmonization of modelling initiatives within model-driven projects should be supported in information provided by the abstraction levels. Other quality features such as suitability, coverage, communication, integration capacities, and mapping support can be analyzed (possibly predicted) by the explicit presence of abstraction levels. Abstraction levels should not have ambiguous concepts. Theorical frameworks such as FRISCO provide definitions about computerized information systems and the *abstraction level zero* through the presence of *processors*. In this way, the lower abstraction level is framed around technogical boundaries where information is processed.

Abstraction levels are a critical approach for understanding information systems and defining the alignment of model-driven initiatives with business, system, or technical scenarios within an IS architecture (in accordance with the MDA specification). The abstraction levels make the use of modelling techniques explicitly, so that a posterior inference process can determines the suitability of the modelling proposal.

The abstraction level challenge includes a discussion about the convenience of the model-driven architecture and its support through the *instanceOf* relation. This relation occurrs between layers, not inside them. No other relations are permissible. This is a constraint artificially imposed without any philosophical or ontological arguments.

The lack of a common consensus about when something is model-driven compliant (challenge 5.2) favors the emergence of self-denominated model-driven initiatives without a formal analysis beyond notational proposals that are supported by a problem context that justifies their formulation. The explicit presence of abstraction levels within a model quality evaluation procedure allows the *convenience* of any model-driven compliance initiative to be taken into account based on the rules and prescripts of each level. For example, decisions about the practical implications for using UML at business levels could be addressed and contrasted against the implications of the model semantics and the scope of the business level.

5.4 Agreeing on a set of generic quality metrics for modelling languages

The applicability of metrics and measurement processes in models has been used to rate specific elements that are associated to model-driven projects. This includes the presence of defects (Marín et al., 2013); the size of diagrams (commonly UML diagrams)(Lange, 2007a); model transformations(van Amstel et al., 2009); metamodels (Monperrus et al., 2008)(Pallec and Dupuy-Chessa, 2013); metamodels with controlled experiments (Yue et al., 2010); etc. Since these application are very specific, works of this kind can only be starting points to define operationalization of specific quality efforts.

Reports about metrics in models present the intention of applying metric approaches derived from software quality works. However, the quality features presented above (Section 2.4) do not include certain associated metrics. The usage and applicability of metrics is highly subjective. Consequently, it is not important to discern which specific field of the model-driven paradigm is the most appropriate to identify and implement metrics (e.g., metrics on notations, metrics for the use of models, metrics on metamodels, metrics for a specific modelling language).

Most of the identified works define metrics for models. We recommend metrics for modelling languages. Some works also define metrics for subsets of languages, e.g., metrics that are specified by metamodelling (López-Fernández et al., 2014). However, the scope of these metrics is limited. Therefore, we consider metrics that can be applicable to any modelling language or sets of modelling languages.

The most important contribution of the metrics should be to consolidate the essential aspects of model management in order to establish a set of core modelling features that can be used. This is a challenge given the large size of the model-driven paradigm compared to traditional software development projects. From a MDE pure viewpoint a set of metrics is required to measure the derived features and issues in the information systems modelling process itself.

Thus, approaches such as Goal-Question-Metric (GQM) or other metric-related techniques can be useful for deriving metrics from the goals associated to the modelling act itself (independent of the degree of subjectivity presented). Modelling goals should be aligned with information systems architectural principles over specific individual considerations derived from the application of specific model-driven approaches.

5.5 Including model transformations in the modelling language quality equation

Model tranformations are critical in model-driven contexts. Modelling languages are often the source or target of model transformations. It is critical to ensure that the modelling languages are appropriate for this purposes.

Transformations constitute the full manifestation of the power of conceptual models in terms of managing the complexity associated with the multiple views and deriving artifacts from the same subject under study. Works such as (Van Amstel, 2010) present new quality features for the transformations. These are derived from a *transformation process* itself (i.e., the rationale of the transformation and its consequences beyond the mere usage of model transformation languages).

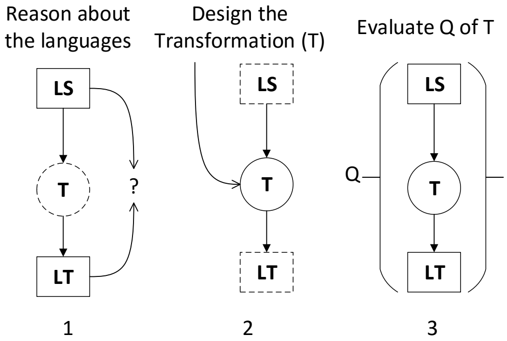


Fig. 8 Proposed order for model transformations.

Some current works propose methods for evaluating the quality of transformation of languages. We think it is important to consider the opposite way, i.e., given the goal of defining a transformation from one modelling perspective to another either horizontally (endogenous) or vertically (exogenous), we need methods to evaluate whether or not the choice of source/target modelling languages is appropriate. This idea is also considered in (da Silva, 2015). The author claims that models must be defined in a consistent and rigorous way; therefore, a *certain level of quality* is required so that models might be properly used in transformation scenarios.

For existing works, a pre-selection of the languages is assumed so that the appropriateness of the transformation is evaluated. However, there are no mechanisms for reasoning whether or not the languages are appropriate. Fig. 8 presents an appropiate order for transformations. It includes reasoning about the languages as the first step, then the design of a transformation, and, finally, the quality evaluation.

The inherent complexity of transformations must be tamed by a process, where the main features of the transformation can be identified and managed. Modelling transformation languages cannot provide full support to phenomena derived from issues such as the following: transformations between modelling languages in the same abstraction level; influence of traceability in the transformation; and implications of information carried in traceability models (Galvão and Goknil, 2007);

addition of information in mappings models; and differences between mapping and transformation models.

Orientations about model transformations as presented in (Mens and Gorp, 2006) consider the mappings and transformation to be a managed process, where activities such as analysis, design, implementation, verification, and deployment can be performed. Both alternatives (mapping and transformation) must be considered in accordance with the MDA principles (the basis for the general consensus around the model-driven initiative).

All decisions about transformations should not be delegated exclusively to the model transformation language employed; it is an artifact of the model transformation process itself. In addition, semantic rules in models (expressed by Object Constraint Languages - OCL - for example) require supporting information about considerations for their translation. Addressing the question about *when the conversion under analysis is a mapping model or transformation model* must be the initial activity and orientation of the process itself.

5.6 Acknowledging the increasing dynamics of models

Taking advantage of the *context of use* of to the *semiotic dimension* of *pragmatics*, in (Barišić et al., 2011), the authors propose the evaluation of the productivity of domain experts through experimental validation of the introduction of DSLs. This is a key issue because it considers the *quality in use* characteristic for DSLs, so quality in model-driven context transcends beyond the *internal quality* presented in (Moody, 2005). Using usability evaluation, the authors provide some traces for the cognitive activities in the context of languages based on user and tasks scenarios.

Unfortunately, experiments of this kind only consider element languages (except the *representation*) as the natural consequence and interface between the syntax, semantics, and users. *A representation must reflect the semantics of the language*, i.e., implicitly the semantics could be derived from the representations. With this term, we considered both diagrams and textual instances of the modelling languages from the perspective of their users.

The MDA revision guide 2.0 promotes this challenge by presenting analytic procedures that are performed once the *data semantics behind the diagrams* are captured[14]. MDA 2.0 prescribes the capture of models in the form of required data for operations such as querying, analyzing, reporting, simulating, and transforming (OMG, 2014).

There is more evidence that models are no longer static representations of realities. The dynamics in models is increasing in MDE environments. By dynamics we refer to interaction with the elements of the model, the navigation through structures of related models, the simulation of behavior (e.g., GUI models), queries on models, etc.

This dynamics is not usually considered by frameworks for the evaluation of quality in modelling languages. However, it is important for the essential management and use of models in MDE projects. Ignoring it can lead to problems in the

[14] The MDA specification particularly promotes the *diagram* term. It can be inferred from previous OMG proposals for managing diagrammatic representations of languages based on arcs and nodes.

final system. Therefore, we believe this challenge must be explicitly considered as part of the quality of (sets of) modelling languages in MDE environments.

Most of the modern proposals about semantics management in model-driven context are too formal and empirical for the community. The lack of an appropriate treatment for representations promotes the presence of modelling tools that do not have the appropiated tools support for modelling purposes (only representations without any association to the semantics). A modelling language can be considered *good* if its associated tool implicitly explains and supports its semantics.

5.7 Streamlining ontological analyses of modelling languages

The reported methods for evaluating quality in models and modelling languages include artifacts such as guidelines, ontological analysis, experimentation, and usability evaluation. Ontological analysis is one of the approach that is most reported to evaluate modelling languages regarding concrete conceptualizations. Works such as (Becker et al., 2010; Opdahl and Henderson-Sellers, 2002; Siau, 2010) give some examples of evaluation processes with the BWW ontological model applied over UML and DSLs respectively. In (Costal et al., 2011), the enhancement of the expressiveness of the UML was proposed based on the analysis using the UFO ontology. The authors in (Ruiz et al., 2014) use an ontological semiotic framework for information systems (FRISCO) as pivot to integrate two modelling languages; ontological elements are used to relate and support the integration between concepts of both languages.

While it is true that ontological guidance provides a powerful tool to help in the understandability of models (Saghafi and Wand, 2014), ontological analysis includes procedures at philosophical levels which may not be accessible (or interesting) for all of the model-driven community. These analyses are performed by method engineers who have a general vision about the implications of modelling languages in model-driven projects. However, most of the model-driven community are final users of modelling languages, so their interests are focused on the applicability of languages in a domain. An *agile* ontological approach is needed to facilitate analysis and reasoning about the applicability of modelling languages, according to the particular characterizations of the domain being modelled.

The term *agile* means the *real knowledge* about the modelling act in accordance with information systems principles. Agile approaches consider constant improvements, short iterations, and the exchange of knowledge and experience among team members (Silva et al., 2015). Current ontological analysis proposals on models and modelling languages limit their application to specific model-driven communities, which are interested in the evaluation of modelling approaches or the promotion of specific modelling proposals. In addition there are several information systems frameworks (not just ontological frameworks) which contribute their own individual conception of information systems. In order to promote ontologic reasonings about modelling implications, we propose an intermediate stage where a native IS neutral description can be used to classify modelling artifacts before starting the inference process with an information system ontology.

Another important advantage that an agile ontological analysis could offer to the model-driven community is its potential use to develop supporting material (orientation, guidelines, etc) for the correct application of modelling-related practices in real contexts. Some examples of practices are the choice of language, adequate usage of tools, management of traceability information in transformation processes, etc.

5.8 Incorporating modelling language quality as a source of technical debt in MDE

Most of the proposed frameworks for quality in models act upon specific model artifacts, abstract syntax, or concrete syntax. These frameworks do not consider the implications of the activities performed in models in terms of the consequences of the good practices that were not followed. This is a critical issue because model-driven projects have the same project constraints as software projects. The only difference is the high abstract level of the project artifacts and the new roles for to domain experts and language users.

The main concern of the term *technical debt* is the consequence of poor software development(Tom et al., 2013). This is a critical issue that is not covered in model-driven processes whose focus is specific operation in models such as model management and model transformations. A landscape for technical debt in software is proposed in (Kruchten et al., 2012) in terms of evolvability and external/internal quality issues. We think that model-driven iniatives cover all the elements of these landscapes since that authors such as (Moody, 2005) suggest models as elements of internal quality software due to their *intermediate nature* in a software development process. Researchers of the Software Engineering Institute (SEI) in (Schmidt) propose a further work that is related to the analysis and management of decisions concerning architecture (expressed as modelling software decisions) because it implies costs, values, and debts for a software development process. The integration between the model-driven engineering and technical debt has not been considered by practicioners of each area despite the enormous potential and benefits for software development processes.

Some of the quality issues reported in Section 3 show concerns about the consequences of model-driven applied practices (especially their formal manifestation as model-driven processes). However, unlike traditional software technical debt, the consequences of MDE activities could cover all of the abstraction levels involved, including business and organizational concerns.

The benefit of considering this challenge is twofold because this implies that model-driven processes must be formulated and formalized. In addition, a prior vision of the consequence of model-driven activities will avoid misalignments with the real application context. Most of the MDE applicability problems are generated by technical incidences in the MDE tools. The consequence of any model-driven activity should be measurable and quantified without waiting until the quality is impacted in an specific scenario.

Technical debt in model-driven contexts has begun to be considered by model-driven practitioners. An example is presented in (Izurieta et al., 2015), where the authors explore the improvement of the software architecture quality through the explicit management of technical debt during modelling tasks. In the opinion of the authors, taking the technical debt into account at modelling levels enhances the value added to MDE and it also promotes the progressive adoption of modelling by regular software practitioners.

6 Conclusions

The virtue of *quality* does not exist per se; it depends on the *subject* under consideration. In MDE contexts there are a plethora of meanings about quality in MDE as consequence of the multiple interpretations about the real scope of the MDE paradigm. This paradigm ranges from between the mere usage of conceptual models to specialized semantic forms. As a relatively young discipline, multiple conceptualizations of quality have not yet been acknowledged by model-driven communities and practitioners. The most critical consequence of this is the reported misalignment between the expectations of real industrial scenarios and the proposals that emerge from academia.

A greater number of quality concepts in model-driven projects have high abstraction level sources with concerns related to the act of modelling itself. Just as there is widespread belief that *good quality models should generate software artifacts with good quality*, there should be a standard conceptualization of the implications of *good quality models*. However, this conceptualization fails because the paradigm does not establish when something is MDE (or is in compliance MDE). For the model-driven case, the significant impact of subjectivity generates multiple efforts and works about the *quality* term, most of which do not address the real expectations, constraints, and requirements of real contexts.

Through two formal literature reviews, we have shown several categories in the definition of quality for the MDE field. We have also analyzed the mismatch of quality evidence between industrial practitioners (and communities of model-driven practioners) and academic researchers. Table 9 and Fig. 9 summarize the main findings of both reviews. Table 9 presents the categories identified in the definition of quality in MDE contexts, classifying them according to their main contribution for evaluation procedures. Sixteen definitions of quality for MDE context were detected. Fig. 9 summarizes the industrial-academic/research mismatch of model-driven quality issues. One hundred twenty-one issues were detected in the works found by grouping explicit statements that show quality problems.

Contribution	Identified category
Specific quality frameworks	Category 1
	Category 2
	Category 3
	Category 13
	Category 14
UML	Category 4
	Category 5
	Category 7
	Category 8
	Category 10
Quality of transformations	Category 6
Concrete scenarios	Category 9
	Category 11
	Category 14
	Category 15
	Category 16

Table 9 Summary of the categories definition about quality in MDE contexts (August 2016).

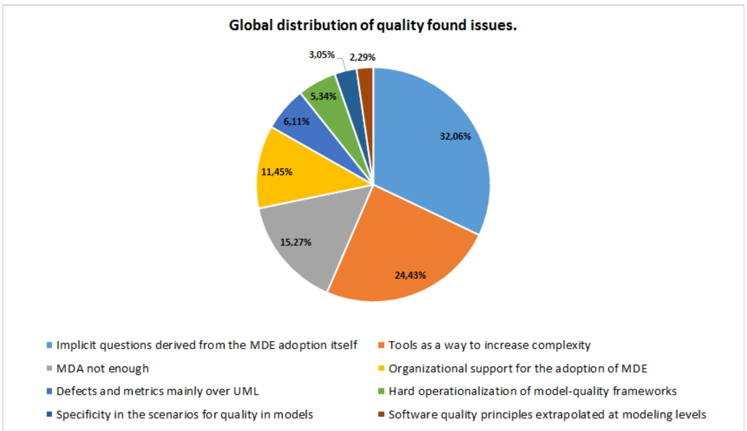


Fig. 9 Global distribution of quality issues in industrial and academic/research contexts.

Detected categories about quality in models are a strong basis from which to start the discussion on this topic. However, most of the MDE core features and challenges are discarded. These include the suitability of languages and their joint usage, the conformity to MDE, the management of abstraction levels, the granularity of models, etc. In Section 4, we showed how these quality issues emerge in real model-driven projects with the modelling only act itself because MDE projects are constrained to business, system, and technical concerns. For this reason, we claim that the model-driven community must pay attention to the challenges formulated in Section 5 in order to derive quality initiatives with an effective impact on the practitioners of the model-driven paradigm, who mostly come from traditional software development contexts.

The diversity of MDE-compliant works and the lack of a general consensus about MDE (possibly similar to the OMG MDA initiative) produce particular definitions about quality. As Krosgtie states: *model quality is still an open issue* (Krogstie, 2012c); it will continue to be an open issue as long as the diversity of ideas about MDE persists. None of the identified categories establish when an artifact can be explicitly considered MDE compliant. Multiple categories confirm that the term *quality in models* does not have a consistent definition and it is defined, conceptualized, and operationalized in different ways depending on the discourse of the previous research proposals (Fettke et al., 2012). Fig. 9 shows the *implicit questions from the adoption of MDE itself* as being the main open issue in the perception of quality in MDE that still does not have a satisfactory response due to the lack of consensus about the scope of the definition of *model-driven compliance.*

The software engineering field has specific standards, efforts, and initiatives that allow practitioners to reach agreements and consensus on the quality conceptualization in software projects. However, the MDE paradigm (which is born from software enginering methods) lacks consensual initiatives due to the multiple new challenges and categories that emerge in quality evaluation in MDE. We believe there must be a comprehensive consensus that takes into account the quality evaluation in MDE by using essential principles of information systems architectures that drive modelling actions and decisions.

Acknowledgements

F.G, would like to thank COLCIENCIAS (Colombia) for funding this work through the Colciencias Grant call 512-2010. This work has been supported by the Generalitat Valenciana Project IDEO (PROMETEOII/2014/039), the European Commission FP7 Project CaaS (611351), and ERDF structural funds.

References

(1985). Iso information processing – documentation symbols and conventions for data, program and system flowcharts, program network charts and system resources charts. *ISO 5807:1985(E)*, pages 1–25.

(2011). Iso/iec/ieee systems and software engineering – architecture description. *ISO/IEC/IEEE 42010:2011(E) (Revision of ISO/IEC 42010:2007 and IEEE Std 1471-2000)*, pages 1–46.

Abran, A., Moore, J. W., Bourque, P., Dupuis, R., and Tripp, L. L. (2013). *Guide to the Software Engineering Body of Knowledge (SWEBOK) version 3 public review*. IEEE. ISO Technical Report ISO/IEC TR 19759.

Agner, L. T. W., Soares, I. W., Stadzisz, P. C., and Simo, J. M. (2013). A brazilian survey on {UML} and model-driven practices for embedded software development. *Journal of Systems and Software*, 86(4):997 – 1005. {SI} : Software Engineering in Brazil: Retrospective and Prospective Views.

Amstel, M. F. V. (2010). The right tool for the right job: Assessing model transformation quality. pages 69–74. Affiliation: Eindhoven University of Technology, P.O. Box 513, 5600 MB, Eindhoven, Netherlands. Cited By (since 1996):1.

Aranda, J., Damian, D., and Borici, A. (2012). Transition to model-driven engineering: What is revolutionary, what remains the same? In *Proceedings of the 15th International Conference on Model Driven Engineering Languages and Systems*, MODELS'12, pages 692–708, Berlin, Heidelberg. Springer-Verlag.

Arendt, T. and Taentzer, G. (2013). A tool environment for quality assurance based on the eclipse modeling framework. *Automated Software Engg.*, 20(2):141–184.

Atkinson, C., Bunse, C., and Wüst, J. (2003). *Driving component-based software development through quality modelling*, volume 2693. Cited By (since 1996):3.

Baker, P., Loh, S., and Weil, F. (2005). Model-driven engineering in a large industrial context motorola case study. In Briand, L. and Williams, C., editors, *Model Driven Engineering Languages and Systems*, volume 3713 of *Lecture Notes in Computer Science*, pages 476–491. Springer Berlin Heidelberg.

Barišić, A., Amaral, V., Goulão, M., and Barroca, B. (2011). Quality in use of domain-specific languages: A case study. In *Proceedings of the 3rd ACM SIGPLAN Workshop on Evaluation and Usability of Programming Languages and Tools*, PLATEAU '11, pages 65–72, New York, NY, USA. ACM.

Becker, J., Bergener, P., Breuker, D., and Rackers, M. (2010). Evaluating the expressiveness of domain specific modeling languages using the bunge-wand-weber ontology. In *System Sciences (HICSS), 2010 43rd Hawaii International Conference on*, pages 1–10.

Bertrand Portier, L. A. (2009). Model driven development misperceptions and challenges.

Bézivin, J. and Kurtev, I. (2005). Model-based technology integration with the technical space concept. In *In: Proceedings of the Metainformatics Symposium, Springer-Verlag*. Springer-Verlag.

Brambilla, M. (2016). How mature is of model-driven engineering as an engineering discipline @ONLINE.

Brambilla, M. and Fraternali, P. (2014). Large-scale model-driven engineering of web user interaction: The webml and webratio experience. *Science of Computer Programming*, 89, Part B(0):71 – 87. Special issue on Success Stories in Model Driven Engineering.

Brown, A. (2009). Simple and practical model driven architecture (mda) @ONLINE.

Bruel, J.-M., Combemale, B., Ober, I., and Raynal, H. (2015). Mde in practice for computational science. *Procedia Computer Science*, 51:660–669.

Budgen, D., Burn, A. J., Brereton, O. P., Kitchenham, B. A., and Pretorius, R. (2011). Empirical evidence about the uml: a systematic literature review. *Software: Practice and Experience*, 41(4):363–392.

Burden, H., Heldal, R., and Whittle, J. (2014). Comparing and contrasting model-driven engineering at three large companies. In *Proceedings of the 8th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*, ESEM '14, pages 14:1–14:10, New York, NY, USA. ACM.

Cabot, J. Has mda been abandoned (by the omg)?

Cabot, J. (2009). Modeling will be commonplace in three years time @ONLINE.

Cachero, C., Poels, G., Calero, C., and Marhuenda, Y. (2007). Towards a Quality-Aware Engineering Process for the Development of Web Applications. Working Papers of Faculty of Economics and Business Administration, Ghent University, Belgium 07/462, Ghent University, Faculty of Economics and Business Administration.

Challenger, M., Kardas, G., and Tekinerdogan, B. (2015). A systematic approach to evaluating domain-specific modeling language environments for multi-agent systems. *Software Quality Journal*, pages 1–41.

Chaudron, M. V., Heijstek, W., and Nugroho, A. (2012). How effective is uml modeling ? *Software & Systems Modeling*, 11(4):571–580. J2: Softw Syst Model.

Chenouard, R., Granvilliers, L., and Soto, R. (2008). Model-driven constraint programming. pages 236–246. Affiliation: CNRS, LINA, Universit de Nantes, France; Affiliation: Pontificia Universidad Catlica de, Valparaiso, Chile. Cited By (since 1996):8.

Clark, T. and Muller, P.-A. (2012). Exploiting model driven technology: a tale of two startups. *Software & Systems Modeling*, 11(4):481–493.

Corneliussen, L. (2008). What do you think of model-driven software development?

Costal, D., Gmez, C., and Guizzardi, G. (2011). Formal semantics and ontological analysis for understanding subsetting, specialization and redefinition of associations in uml. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 6998 LNCS:189–203. cited By (since 1996)3.

Cruz-Lemus, J. A., Maes, A., Génous, M., Poels, G., and Piattini, M. (2010). The impact of structural complexity on the understandability of uml statechart diagrams. *Information Sciences*, 180(11):2209–2220. Cited By (since 1996):14.

Cuadrado, J. S., Izquierdo, J. L. C., and Molina, J. G. (2014). Applying model-driven engineering in small software enterprises. *Science of Computer Programming*, 89, Part B(0):176 – 198. Special issue on Success Stories in Model Driven Engineering.

da Silva, A. R. (2015). Model-driven engineering: A survey supported by the unified conceptual model. *Computer Languages, Systems & Structures*, 43:139 – 155.

da Silva Teixeira, d. G. M., Quirino, G. K., Gailly, F., de Almeida Falbo, R., Guizzardi, G., and Perini Barcellos, M. (2016). *PoN-S: A Systematic Approach for Applying the Physics of Notation (PoN)*, pages 432–447. Springer International Publishing, Cham.

Davies, I., Green, P., Rosemann, M., Indulska, M., and Gallo, S. (2006). How do practitioners use conceptual modeling in practice? *Data & Knowledge Engineering*, 58(3):358 – 380. Including the special issue : {ER} 2004ER 2004.

Davies, J., Milward, D., Wang, C.-W., and Welch, J. (2015). Formal model-driven engineering of critical information systems. *Science of Computer Programming*, 103(0):88 – 113. Selected papers from the First International Workshop on Formal Techniques for Safety-Critical Systems (FTSCS 2012).

de Oca, I. M.-M., Snoeck, M., Reijers, H. A., and Rodrguez-Morffi, A. (2015). A systematic literature review of studies on business process modeling quality. *Information and Software Technology*, 58:187 – 205.

DenHaan, J. (2009). 8 reasons why model driven development is dangerous @ONLINE.

DenHaan, J. (2010). Model driven engineering vs the commando pattern @ONLINE.

DenHaan, J. (2011a). Why aren't we all doing model driven development yet @ONLINE.

DenHaan, J. (2011b). Why there is no future model driven development @ONLINE.

Di Ruscio, D., Iovino, L., and Pierantonio, A. (2013). Managing the coupled evolution of metamodels and textual concrete syntax specifications. pages 114–121. cited By (since 1996)0.

Dijkman, R. M., Dumas, M., and Ouyang, C. (2008). Semantics and analysis of business process models in {BPMN}. *Information and Software Technology*, 50(12):1281 – 1294.

Domínguez-Mayo, F. J., Escalona, M. J., Mejías, M., Ramos, I., and Fernández, L. (2011). A framework for the quality evaluation of mdwe methodologies and information technology infrastructures. *International Journal of Human Capital and Information Technology Professionals*, 2(4):11–22.

Domínguez-Mayo, F. J., Escalona, M. J., Mejías, M., and Torres, A. H. (2010). A quality model in a quality evaluation framework for mdwe methodologies. pages 495–506. Affiliation: Departamento de Lenguajes y Sistemas Informíticos, University of Seville, Seville, Spain. Cited By (since 1996):1.

Dubray, J.-J. (2011). Why did mde miss the boat?

Escalona, M. J., Gutiérrez, J. J., Pérez-Pérez, M., Molina, A., Domínguez-Mayo, E., and Domínguez-Mayo, F. J. (2011). *Measuring the Quality of Model-Driven Projects with NDT-Quality*, pages 307–317. Springer New York.

Espinilla, M., Domínguez-Mayo, F. J., Escalona, M. J., Mejías, M., Ross, M., and Staples, G. (2011). *A Method Based on AHP to Define the Quality Model of QuEF*, volume 123, pages 685–694. Springer Berlin Heidelberg.

Fabra, J., Castro, V. D., lvarez, P., and Marcos, E. (2012). Automatic execution of business process models: Exploiting the benefits of model-driven engineering approaches. *Journal of Systems and Software*, 85(3):607 – 625. Novel approaches in the design and implementation of systems/software architecture.

Falkenberg, E. D., Hesse, W., Lindgreen, P., Nilsson, B. E., Oei, J. L. H., Rolland, C., Stamper, R. K., Assche, F. J. M. V., Verrijn-Stuart, A. A., and Voss, K. (1996). Frisco: A framework of information system concepts. Technical report, The IFIP WG 8. 1 Task Group FRISCO.

Fettke, P., Houy, C., Vella, A.-L., and Loos, P. (2012). *Towards the Reconstruction and Evaluation of Conceptual Model Quality Discourses Methodical Framework and Application in the Context of Model Understandability*, volume 113 of *Lecture Notes in Business Information Processing*, chapter 28, pages 406–421. Springer Berlin Heidelberg.

Finnie, S. (2015). Modeling community: Are we missing something?

Fournier, C. (2008). Is uml practical? @ONLINE.

France, R. and Rumpe, B. (2007). Model-driven development of complex software: A research roadmap. In *Future of Software Engineering, 2007. FOSE '07*, pages 37–54.

Gallego, M., Giraldo, F. D., and Hitpass, B. (2015). Adapting the pbec-otss software selection approach for bpm suites: an application case. In *2015 34th International Conference of the Chilean Computer Science Society (SCCC)*, pages 1–10.

Galvão, I. and Goknil, A. (2007). Survey of traceability approaches in model-driven engineering. pages 313–324. cited By (since 1996)22.

Giraldo, F., España, S., Giraldo, W., and Pastor, O. (2015). Modelling language quality evaluation in model-driven information systems engineering: A roadmap. In *Research Challenges in Information Science (RCIS), 2015 IEEE 9th International Conference on*, pages 64–69.

Giraldo, F., España, S., and Pastor, O. (2014). Analysing the concept of quality in model-driven engineering literature: A systematic review. In *Research Challenges in Information Science (RCIS), 2014 IEEE Eighth International Conference on*, pages 1–12.

Giraldo, F. D., España, S., and Pastor, O. (2016). Evidences of the mismatch between industry and academy on modelling language quality evaluation. *CoRR*, abs/1606.02025.

González, C. and Cabot, J. (2014). Formal verification of static software models in mde: A systematic review. *Information and Software Technology*, 56(8):821–838. cited By (since 1996)0.

González, C. A., Büttner, F., Clarisó, R., and Cabot, J. (2012). Emftocsp: A tool for the lightweight verification of emf models. pages 44–50. Affiliation: cole des Mines de Nantes, INRIA, LINA, Nantes, France; Affiliation: Universitat Oberta de Catalunya, Barcelona, Spain. Cited By (since 1996):1.

Gorschek, T., Tempero, E., and Angelis, L. (2014). On the use of software design models in software development practice: An empirical investigation. *Journal of Systems and Software*, 95(0):176 – 193.

Goulão, M., Amaral, V., and Mernik, M. (2016). Quality in model-driven engineering: a tertiary study. *Software Quality Journal*, pages 1–33.

Grobshtein, Y. and Dori, D. (2011). Generating sysml views from an opm model: Design and evaluation. *Systems Engineering*, 14(3):327–340.

Haan, J. d. (2008). 8 reasons why model-driven approaches (will) fail.

Harel, D. and Rumpe, B. (2000). Modeling languages: Syntax, semantics and all that stuff, part i: The basic stuff. Technical report, Jerusalem, Israel, Israel.

Harel, D. and Rumpe, B. (2004). Meaningful modeling: What's the semantics of semantics? *Computer*, 37(10):64–72.

Hebig, R. and Bendraou, R. (2014). On the need to study the impact of model driven engineering on software processes. In *Proceedings of the 2014 International Conference on Software and System Process*, ICSSP 2014, pages 164–168, New York, NY, USA. ACM.

Heidari, F. and Loucopoulos, P. (2014). Quality evaluation framework (qef): Modeling and evaluating quality of business processes. *International Journal of Accounting Information Systems*, 15(3):193 – 223. Business Process Modeling.

Heymans, P., Schobbens, P. Y., Trigaux, J. C., Bontemps, Y., Matulevicius, R., and Classen, A. (2008). Evaluating formal properties of feature diagram languages. *Software, IET*, 2(3):281–302. ID: 2.

Hindawi, M., Morel, L., Aubry, R., and Sourrouille, J.-L. (2009). *Description and Implementation of a UML Style Guide*, volume 5421, pages 291–302. Springer Berlin Heidelberg.

Hoang, D. (2012). Current limitations of mdd and its implications @ONLINE.

Hodges, W. (2013). Model theory. In Zalta, E. N., editor, *The Stanford Encyclopedia of Philosophy*. Fall 2013 edition.

Hutchinson, J., Rouncefield, M., and Whittle, J. (2011a). Model-driven engineering practices in industry. In *Proceedings of the 33rd International Conference on Software Engineering*, ICSE '11, pages 633–642, New York, NY, USA. ACM.

Hutchinson, J., Whittle, J., and Rouncefield, M. (2014). Model-driven engineering practices in industry: Social, organizational and managerial factors that lead to success or failure. *Science of Computer Programming*, 89, Part B(0):144 – 161. Special issue on Success Stories in Model Driven Engineering.

Hutchinson, J., Whittle, J., Rouncefield, M., and Kristoffersen, S. (2011b). Empirical assessment of mde in industry. In *Proceedings of the 33rd International Conference on Software Engineering*, ICSE '11, pages 471–480, New York, NY, USA. ACM.

Igarza, I. M. H., Boada, D. H. G., and Valdés, A. P. (2012). Una introducción al desarrollo de software dirigido por modelos. *Serie Científica*, 5(3).

ISO/IEC (2001). *ISO/IEC 9126. Software engineering – Product quality*. ISO/IEC.

Izurieta, C., Rojas, G., and Griffith, I. (2015). Preemptive management of model driven technical debt for improving software quality. In *Proceedings of the 11th International ACM SIGSOFT Conference on Quality of Software Architectures*, QoSA '15, pages 31–36, New York, NY, USA. ACM.

Jalali, S. and Wohlin, C. (2012). Systematic literature studies: Database searches vs. backward snowballing. In *Proceedings of the ACM-IEEE International Symposium on Empirical Software Engineering and Measurement*, ESEM '12, pages 29–38, New York, NY, USA. ACM.

Kahraman, G. and Bilgen, S. (2013). A framework for qualitative assessment of domain-specific languages. *Software & Systems Modeling*, pages 1–22.

Kessentini, M., Langer, P., and Wimmer, M. (2013). Searching models, modeling search: On the synergies of sbse and mde. In *Combining Modelling and Search-Based Software Engineering (CMSBSE), 2013 1st International Workshop on*, pages 51–54.

Kitchenham, B. and Charters, S. (2007). Guidelines for performing Systematic Literature Reviews in Software Engineering. Technical Report EBSE 2007-001, Keele University and Durham University Joint Report.

Kitchenham, B., Pfleeger, S., Pickard, L., Jones, P., Hoaglin, D., El Emam, K., and Rosenberg, J. (2002). Preliminary guidelines for empirical research in software engineering. *Software Engineering, IEEE Transactions on*, 28(8):721–734.

Klinke, M. (2008). Do you use mda/mdd/mdsd, any kind of model-driven approach? will it be the future?

Köhnlein, J. (2013). Eclipse diagram editors from a user's perspective.

Kolovos, D. S., Paige, R. F., and Polack, F. A. (2008). The grand challenge of scalability for model driven engineering. In *Models in Software Engineering*, pages 48–53. Springer.

Kolovos, D. S., Rose, L. M., Matragkas, N., Paige, R. F., Guerra, E., Cuadrado, J. S., De Lara, J., Ráth, I., Varró, D., Tisi, M., and Cabot, J. (2013). A research roadmap towards achieving scalability in model driven engineering. In *Proceedings of the Workshop on Scalability in Model Driven Engineering*, BigMDE '13, pages 2:1–2:10, New York, NY, USA. ACM.

Krill, P. (2016). Uml to be ejected from microsoft visual studio (infoworld).

Krogstie, J. (2012a). *Model-Based Development and Evolution of Information Systems: A Quality Approach*. Springer Publishing Company, Incorporated.

Krogstie, J. (2012b). *Quality of Modelling Languages*, pages 249–280. Springer London, London.

Krogstie, J. (2012c). *Quality of Models*, pages 205–247. Springer London.

Krogstie, J. (2012d). *Specialisations of SEQUAL*, pages 281–326. Springer London.

Krogstie, J., Lindland, O. I., and Sindre, G. (1995). Defining quality aspects for conceptual models. In *Proceedings of the IFIP International Working Conference on Information System Concepts: Towards a Consolidation of Views*, pages 216–231, London, UK, UK. Chapman & Hall, Ltd.

Kruchten, P. (2000). *The Rational Unified Process: An Introduction, Second Edition*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2nd edition.

Kruchten, P., Nord, R., and Ozkaya, I. (2012). Technical debt: From metaphor to theory and practice. *Software, IEEE*, 29(6):18–21.

Kulkarni, V., Reddy, S., and Rajbhoj, A. (2010). Scaling up model driven engineering experience and lessons learnt. In Petriu, D., Rouquette, N., and Haugen, y., editors, *Model Driven Engineering Languages and Systems*, volume 6395 of *Lecture Notes in Computer Science*, pages 331–345. Springer Berlin Heidelberg.

Laguna, M. A. and Marqués, J. M. (2010). Uml support for designing software product lines: The package merge mechanism. 16(17):2313–2332.

Lange, C. (2007a). Model size matters. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 4364 LNCS:211–216. cited By (since 1996)1.

Lange, C. and Chaudron, M. (2005). Managing Model Quality in UML-Based Software Development. In *Software Technology and Engineering Practice, 2005. 13th IEEE International Workshop on*, pages 7–16.

Lange, C., Chaudron, M. R. V., Muskens, J., Somers, L. J., and Dortmans, H. M. (2003). An empirical investigation in quantifying inconsistency and incompleteness of uml designs. In *Incompleteness of UML Designs, Proc. Workshop on Consistency Problems in UML-based Software Development, 6 th International Conference on Unified Modeling Language, UML 2003*.

Lange, C., DuBois, B., Chaudron, M., and Demeyer, S. (2006). An experimental investigation of uml modeling conventions. In Nierstrasz, O., Whittle, J., Harel, D., and Reggio, G., editors, *Model Driven Engineering Languages and Systems*, volume 4199 of *Lecture Notes in Computer Science*, pages 27–41. Springer Berlin Heidelberg.

Lange, C. F. J. and Chaudron, M. R. V. (2006). Effects of defects in uml models: an experimental investigation. In *Proceedings of the 28th international conference on Software engineering*, ICSE '06, pages 401–411, New York, NY, USA. ACM.

Lange, C. J. (2007b). *Model Size Matters*, volume 4364, pages 211–216. Springer Berlin Heidelberg.

Laurent, Y., Bendraou, R., and Gervais, M. P. (2013). Executing and debugging uml models: An fuml extension. pages 1095–1102. Affiliation: LIP6, UPMC Paris Universitas, France; Affiliation: LIP6, University of Paris Ouest, Nanterre, France.

Le Pallec, X. and Dupuy-Chessa, S. (2013). Support for quality metrics in metamodelling. In *Proceedings of the Second Workshop on Graphical Modeling Language Development*, GMLD '13, pages 23–31, New York, NY, USA. ACM.

Linders, B. (2015). New developments in model driven software engineering.

Lindland, O. I., Sindre, G., and Sølvberg, A. (1994). Understanding quality in conceptual modeling. *IEEE Softw.*, 11(2):42–49.

López-Fernández, J. J., Guerra, E., and de Lara, J. (2014). Assessing the quality of meta-models. *11th Workshop on Model Driven Engineering, Verification and Validation MoDeVVa 2014*, page 10.

Lukman, T., Godena, G., Gray, J., Heriko, M., and Strmnik, S. (2013). Model-driven engineering of process control software beyond device-centric abstractions. *Control Engineering Practice*, 21(8):1078 – 1096.

Maes, A. and Poels, G. (2007). Evaluating quality of conceptual modelling scripts based on user perceptions. *Data Knowl. Eng.*, 63(3):701–724.

Marín, B., Giachetti, G., Pastor, O., and Abran, A. (2010). A quality model for conceptual models of mdd environments. *Adv. Soft. Eng.*, 2010:1:1–1:17.

Marín, B., Giachetti, G., Pastor, O., Vos, T. E. J., and Abran, A. (2013). Using a functional size measurement procedure to evaluate the quality of models in mdd environments. *ACM Trans. Softw. Eng. Methodol.*, 22(3):26:1–26:31.

Marín, B., Salinas, A., Morandé, J., Giachetti, G., and de la Vara, J. (2014). Key features for a successful model-driven development tool. In *Model-Driven Engineering and Software Development (MODELSWARD), 2014 2nd International Conference on*, pages 541–548. IEEE.

Matinlassi, M. (2005). Quality-driven software architecture model transformation. In *Software Architecture, 2005. WICSA 2005. 5th Working IEEE/IFIP Conference on*, pages 199–200.

Mayerhofer, T. (2012). Testing and debugging uml models based on fuml. In *Software Engineering (ICSE), 2012 34th International Conference on*, pages 1579–1582. ID: 7.

Mens, T. and Gorp, P. V. (2006). A taxonomy of model transformation. *Electronic Notes in Theoretical Computer Science*, 152(0):125 – 142. Proceedings of the International Workshop on Graph and Model Transformation (GraMoT 2005) Graph and Model Transformation 2005.

Merilinna, J. (2005). *A Tool for Quality-Driven Architecture Model Transformation*. PhD thesis, VTT Technical Research Centre of Finland.

Mijatov, S., Langer, P., and Mayerhofer, T. (2013). A framework for testing uml activities based on fuml. In *Workshop on Model Driven Engineering, Verification and Validation - MoDeVVa 2013*, volume 1069/, pages 11–20. CEUR.

Mohagheghi, P. and Aagedal, J. (2007). Evaluating quality in model-driven engineering. In *Proceedings of the International Workshop on Modeling in Software Engineering*, MISE '07, pages 6–, Washington, DC, USA. IEEE Computer Society.

Mohagheghi, P. and Dehlen, V. (2008a). *Developing a quality framework for model-driven engineering*, volume 5002 LNCS. Cited By (since 1996):4.

Mohagheghi, P. and Dehlen, V. (2008b). Where is the proof? - a review of experiences from applying mde in industry. In Schieferdecker, I. and Hartman, A., editors, *Model Driven Architecture Foundations and Applications*, volume 5095 of *Lecture Notes in Computer Science*, pages 432–443. Springer Berlin Heidelberg.

Mohagheghi, P., Dehlen, V., and Neple, T. (2009a). Definitions and approaches to model quality in model-based software development - a review of literature. *Inf. Softw. Technol.*, 51(12):1646–1669.

Mohagheghi, P., Fernandez, M., Martell, J., Fritzsche, M., and Gilani, W. (2009b). Mde adoption in industry: Challenges and success criteria. In Chaudron, M., editor, *Models in Software Engineering*, volume 5421 of *Lecture Notes in Computer Science*, pages 54–59. Springer Berlin Heidelberg.

Mohagheghi, P., Gilani, W., Stefanescu, A., and Fernandez, M. (2013a). An empirical study of the state of the practice and acceptance of model-driven engineering in four industrial cases. *Empirical Software Engineering*, 18(1):89–116.

Mohagheghi, P., Gilani, W., Stefanescu, A., Fernandez, M., Nordmoen, B., and Fritzsche, M. (2013b). Where does model-driven engineering help? experiences from three industrial cases. *Software and Systems Modeling*, 12(3):619–639. cited By (since 1996)0.

Molina, F. and Toval, A. (2009). Integrating usability requirements that can be evaluated in design time into model driven engineering of web information systems. *Advances in Engineering Software*, 40(12):1306 – 1317. Designing, modelling and implementing interactive systems.

Monperrus, M., Jzquel, J.-M., Champeau, J., and Hoeltzener, B. (2008). *Measuring models.* cited By (since 1996)4.

Moody, D. (2006). Dealing with "map shock": A systematic approach for managing complexity in requirements modelling. Luxembourg.

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.

Moody, D. L. (2009). The ´physics´ of notations: Toward a scientific basis for constructing visual notations in software engineering. *IEEE Transactions on Software Engineering*, 35(6):756–779.

Moody, D. L. and Shanks, G. G. (2003). Improving the quality of data models: empirical validation of a quality management framework. *Inf. Syst.*, 28(6):619–650.

Moody, D. L., Sindre, G., Brasethvik, T., and Sølvberg, A. (2002). Evaluating the quality of process models: Empirical testing of a quality framework. In *Proceedings of the 21st International Conference on Conceptual Modeling*, ER '02, pages 380–396, London, UK, UK. Springer-Verlag.

Mora, B., Ruiz, F., García, F., and Piattini, M. (2006). Definición de lenguajes de modelos mda vs dsl.

Morais, F. and da Silva, A. R. (2015). Assessing the quality of user-interface modeling languages. In *Proceedings of the 17th International Conference on Enterprise Information Systems*, pages 311–319.

Mussbacher, G., Amyot, D., Breu, R., Bruel, J.-M., Cheng, B., Collet, P., Combemale, B., France, R., Heldal, R., Hill, J., Kienzle, J., Schttle, M., Steimann, F., Stikkolorum, D., and Whittle, J. (2014). The relevance of model-driven engineering thirty years from now. In Dingel, J., Schulte, W., Ramos, I., Abraho, S., and Insfran, E., editors, *Model-Driven Engineering Languages and Systems*, volume 8767 of *Lecture Notes in Computer Science*, pages 183–200. Springer International Publishing.

Nelson, H. J., Poels, G., Genero, M., and Piattini, M. (2005). Quality in conceptual modeling: Five examples of the state of the art. *Data Knowl. Eng.*, 55(3):237–242.

Nelson, H. J., Poels, G., Genero, M., and Piattini, M. (2012). A conceptual modeling quality framework. *Software Quality Journal*, 20(1):201–228.

Nugroho, A. (2009). Level of detail in {UML} models and its impact on model comprehension: A controlled experiment. *Information and Software Technology*, 51(12):1670 – 1685. Quality of {UML} Models.

OMG (2003). MDA Guide Version 1.0.1.

OMG (2014). MDA Guide revision 2.0.

OMG (2016). Ea-mde: What affects the success of mde in industry @ONLINE.

Opdahl, A. L. and Henderson-Sellers, B. (2002). Ontological evaluation of the uml using the bungewandweber model. *Software and Systems Modeling*, 1(1):43–67.

Ortiz, J. C., Quinteros, E., Abuawad, O., Torricio, F., and Ojeda, J. D. (2013). Primer parcial-mda (model driven architecture) @ONLINE.

Pallec, X. and Dupuy-Chessa, S. (2013). Support for quality metrics in metamodelling. pages 23–31. cited By (since 1996)0.

Panach, J. I., España, S., Dieste, Ó., Pastor, Ó., and Juristo, N. (2015a). In search of evidence for model-driven development claims: An experiment on quality, effort, productivity and satisfaction. *Information and Software Technology*, 62:164 – 186.

Panach, J. I., Juristo, N., Valverde, F., and Pastor, Ó. (2015b). A framework to identify primitives that represent usability within model-driven development methods. *Information and Software Technology*, 58(0):338 – 354.

Petre, M. (2013). Uml in practice. In *Proceedings of the 2013 International Conference on Software Engineering*, ICSE '13, pages 722–731, Piscataway, NJ, USA. IEEE Press.

Piattini, M., Poels, G., Genero, M., Fernández-Saez, A. M., and Nelson, H. J. (2011). Research review: A systematic literature review on the quality of uml models. *J. Database Manage.*, 22(3):46–70.

Picek, R. and Strahonja, V. (2007). Model driven development-future or failure of software development. In *IIS*, volume 7, pages 407–413.

Pierson, H. (2007). Model-driven development (part 2) @ONLINE.

Planas, E., Cabot, J., and Gómez, C. (2016). Lightweight and static verification of {UML} executable models. *Computer Languages, Systems & Structures*, 46:66 – 90.

Platania, G. (2016). Model driven architecture don't work! @ONLINE.

Poruban, J., Bacikova, M., Chodarev, S., and Nosal, M. (2014). Pragmatic model-driven software development from the viewpoint of a programmer: Teaching experience. In *Computer Science and Information Systems (FedCSIS), 2014 Federated Conference on*, pages 1647–1656. IEEE.

Quintero, J., Rucinque, P., Anaya, R., and Piedrahita, G. (2012). How face the top mde adoption problems. In *Informatica (CLEI), 2012 XXXVIII Conferencia Latinoamericana En*, pages 1–10. IEEE.

Quintero, J. B. and Muñoz, J. F. D. (2011). Reflexiones acerca de la adopción de enfoques centrados en modelos en el desarrollo de software. *Ingenieria y Universidad*, 15(1):219–243.

Quora (2014). Is uml trivial? @ONLINE.

Quora (2015a). Is the uml still widely used? is it still an important tool in today's industry?@ONLINE.

Quora (2015b). Why has uml usage declined in industry? @ONLINE.

Reijers, H. A., Mendling, J., and Recker, J. (2015). *Business Process Quality Management*, pages 167–185. Springer Berlin Heidelberg, Berlin, Heidelberg.

Rios, E., Bozheva, T., Bediaga, A., and Guilloreau, N. (2006). Mdd maturity model: A roadmap for introducing model-driven development. In Rensink, A. and Warmer, J., editors, *Model Driven Architecture Foundations and Applications*, volume 4066 of *Lecture Notes in Computer Science*, pages 78–89. Springer Berlin Heidelberg.

Ruiz, M., Costal, D., España, S., Franch, X., and Pastor, Ó. (2014). Integrating the goal and business process perspectives in information system analysis. In Jarke, M., Mylopoulos, J., Quix, C., Rolland, C., Manolopoulos, Y., Mouratidis, H., and Horkoff, J., editors, *Advanced Information Systems Engineering*, volume 8484 of *Lecture Notes in Computer Science*, pages 332–346. Springer International Publishing.

Saghafi, A. and Wand, Y. (2014). Do ontological guidelines improve understandability of conceptual models? a meta-analysis of empirical work. In *System Sciences (HICSS), 2014 47th Hawaii International Conference on*, pages 4609–4618.

Sayeb, K., RIEU, D., Mandran, N., and Dupuy-Chessa., S. (2012). Qualité des langages de modélisation et des modèles : vers un catalogue des patrons collaboratifs. pages 429–446, Montpellier, France.

Schalles, C. (2013). *A Framework for Usability Evaluation of Modeling Languages (FUEML)*, chapter 4, pages 43–68. Springer Fachmedien Wiesbaden.

Schmidt, D. C. Strategic management of architectural technical debt (on-line).

Seddon, P. B. (1997). A respecification and extension of the delone and mclean model of is success. *Information Systems Research*, 8(3):240–253.

Shekhovtsov, V. A., Mayr, H. C., and Kop, C. (2014). Chapter 3 - harmonizing the quality view of stakeholders. In Stal, I. M. B. E. R., editor, *Relating System Quality and Software Architecture*, pages 41 – 73. Morgan Kaufmann, Boston.

Siau, K. (2010). An analysis of unified modeling language (uml) graphical constructs based on bww ontology. *Journal of Database Management*, 21(1):i–viii. cited By (since 1996)2.

Silva, F. S., Soares, F. S. F., Peres, A. L., de Azevedo, I. M., Vasconcelos, A. P. L., Kamei, F. K., and de Lemos Meira, S. R. (2015). Using {CMMI} together with agile software development: A systematic review. *Information and Software Technology*, 58(0):20 – 43.

Singh, Y. and Sood, M. (2009). Model driven architecture: A perspective. In *Advance Computing Conference, 2009. IACC 2009. IEEE International*, pages 1644–1652. IEEE.

Staron, M. (2006). Adopting model driven software development in industry a case study at two companies. In Nierstrasz, O., Whittle, J., Harel, D., and Reggio, G., editors, *Model Driven Engineering Languages and Systems*, volume 4199 of *Lecture Notes in Computer Science*, pages 57–72. Springer Berlin Heidelberg.

Störrle, H. and Fish, A. (2013). Towards an operationalization of the physics of notations for the analysis of visual languages. In Moreira, A., Schtz, B., Gray, J., Vallecillo, A., and Clarke, P., editors, *Model-Driven Engineering Languages and Systems*, volume 8107 of *Lecture Notes in Computer Science*, pages 104–120. Springer Berlin Heidelberg.

Tairas, R. and Cabot, J. (2013). Corpus-based analysis of domain-specific languages. *Software & Systems Modeling*, pages 1–16.

Teppola, S., Parviainen, P., and Takalo, J. (2009). Challenges in deployment of model driven development. In *Software Engineering Advances, 2009. ICSEA '09. Fourth International Conference on*, pages 15–20.

Thompson, S. K. and Seber, G. A. F. (1996). *Adaptive Sampling*. John Wiley & Sons, New York, USA, 1st edition.

Tom, E., Aurum, A., and Vidgen, R. (2013). An exploration of technical debt. *Journal of Systems and Software*, 86(6):1498 – 1516.

Tomassetti, F., Torchiano, M., Tiso, A., Ricca, F., and Reggio, G. (2012). Maturity of software modelling and model driven engineering: A survey in the italian industry. In *Evaluation Assessment in Software Engineering (EASE 2012), 16th International Conference on*, pages 91–100.

Tone (2010). What are the benefits and risks of moving to a model driven architecture approach?

Torchiano, M., Tomassetti, F., Ricca, F., Tiso, A., and Reggio, G. (2013). Relevance, benefits, and problems of software modelling and model driven techniquesa survey in the italian industry. *Journal of Systems and Software*, 86(8):2110 – 2126.

Vallecillo, A. (2010). *On the Combination of Domain Specific Modeling Languages Modelling Foundations and Applications*, volume 6138 of *Lecture Notes in Computer Science*, pages 305–320. Springer Berlin / Heidelberg.

Vallecillo, A. (2014). On the industrial adoption of model driven engineering. is your company ready for mde? *International Journal of Information Systems and Software Engineering for Big Companies*.

Van Amstel, M. (2010). The right tool for the right job: Assessing model transformation quality. pages 69–74. cited By (since 1996)3.

van Amstel, M., Lange, C., and van den Brand, M. (2009). Using metrics for assessing the quality of asf+sdf model transformations. In Paige, R., editor, *Theory and Practice of Model Transformations*, volume 5563 of *Lecture Notes in Computer Science*, pages 239–248. Springer Berlin Heidelberg.

Van Der Straeten, R., Mens, T., and Van Baelen, S. (2009). Challenges in model-driven software engineering. In Chaudron, M., editor, *Models in Software Engineering*, volume 5421 of *Lecture Notes in Computer Science*, pages 35–47. Springer Berlin Heidelberg.

Vara, J. M. and Marcos, E. (2012). A framework for model-driven development of information systems: Technical decisions and lessons learned. *Journal of Systems and Software*, 85(10):2368 – 2384. Automated Software Evolution.

Wehrmeister, M. A., de Freitas, E. P., Binotto, A. P. D., and Pereira, C. E. (2014). Combining aspects and object-orientation in model-driven engineering for distributed industrial mechatronics systems. *Mechatronics*, 24(7):844 – 865. 1. Model-Based Mechatronic System Design 2. Model Based Engineering.

Whittle, J., Hutchinson, J., and Rouncefield, M. (2014). The state of practice in model-driven engineering. *Software, IEEE*, 31(3):79–85.

Whittle, J., Hutchinson, J., Rouncefield, M., Burden, H., and Heldal, R. (2013). Industrial adoption of model-driven engineering: Are the tools really the problem? In Moreira, A., Schtz, B., Gray, J., Vallecillo, A., and Clarke, P., editors, *Model-Driven Engineering Languages and Systems*, volume 8107 of *Lecture Notes in Computer Science*, pages 1–17. Springer Berlin Heidelberg.

Whittle, J., Hutchinson, J., Rouncefield, M., Burden, H., and Heldal, R. (2015). A taxonomy of tool-related issues affecting the adoption of model-driven engineering. *Software & Systems Modeling*, pages 1–19.

Wohlin, C. (2014). Guidelines for snowballing in systematic literature studies and a replication in software engineering. In *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering*, EASE '14, pages 38:1–38:10, New York, NY, USA. ACM.

Yue, T., Ali, S., and Elaasar, M. (2010). A framework for measuring quality of models: Experiences from a series of controlled experiments. Technical Report 2010-17 (v2), Simula Research Laboratory.

Fáber D. Giraldo is a System and Computer Engineer from the University of Quindío, Colombia (with a grant from the Ministry of Education of Colombia). He has a Ms.Eng. degree with emphasis on Informatics from EAFIT University, Colombia (with a grant from EAFIT University). He holds a Ph.D.(C) in Informatics from the Universidad Politécnica de Valencia, Spain (with a grant from the National administrative department of Science, Technology and Innovation of Colombia - COLCIENCIAS) and is currently working with the PROS Research Center. He is a full assistant professor in the Faculty of Engineering at the University of Quindío, He coordinates the Master in Engineering with emphasis on Software Engineering at the University of Quindío. He is a researcher of SINFOCI (by its acronym in spanish) group and is recognized as Associate Researcher type I by COLCIENCIAS. His research interests include software engineering, model-driven engineering, software quality, quality in model-driven engineering, software architecture, enterprise architecture and HCI.

Sergio España is a full-time lecturer (docent) at Utrecht University, where he teaches in the Bachelor in Information Science (Informatikunde) and the Master in Business Informatics programs. He is a member of the Organization and Information research group, where he leads the Responsible Software research line. He holds a PhD in Computer Science (2011) from Universitat Politècnica de València (Spain). He has participated in international applied research projects (e.g., ITEA2 UsiXML, FP7 CaaS). He has published in relevant requirements engineering (RE) and conceptual modelling conferences and journals (e.g., RE, ER, CAiSE, INTERACT, J.UCS, Informatik-Spektrum, BISE, IST, Inf. Syst.). He has contributed as Programme Committee (PC) chair of RCIS 2016, PoEM 2015, CLEI Software Engineering Symposium (2014, 2013), and PC member of international conferences and workshops (e.g, CAiSE, HWID, ONTOSE, VORTE).

Óscar Pastor is a full professor and director of the Research Centre on *Software Production Methods* (PROS) at the Universitat Politècnica de València (Spain). He received his Ph.D. in 1992. He was a researcher at HP Labs, Bristol, UK. He has published more than two hundred research papers in conference proceedings, journals, and books. He has received numerous research grants from public institutions and private industry and has been keynote speaker at several conferences and workshops. As chair of the ER Steering Committee and member of the SC of conferences such as CAiSE, ESEM, ICWE, CIbSE, or RCIS, his research activities focus on conceptual modelling, web engineering, requirements engineering, information systems, model-based software production, and genomic information systems.

William J. Giraldo is an Electrical Engineer from the University of Valle (Colombia). He holds the following degrees: Master in Automatics, University of Valle (Colombia); Master in Advanced Informatics, Castilla la Mancha University (Spain); PhD in Informatics, Castilla la Mancha University (Spain). He is currently an associate professor at the Faculty of Engineering of the University of Quindío (Colombia) and leader of the Information Systems and Industrial Control research group (SINFOCI) at the University of Quindío.

A A multiple modelling languages quality scenario

The following scenario is based on a real project from the University of Quindío (Colombia); the implementation of an information system for institutional academic quality management. This system includes all the resources, processes, technology platforms, and legal frameworks required to achieve the *institutional quality accreditation* certification, which is awarded by the Ministry of Education in Colombia to universities that demonstrate excellence in the exercise of their academic and research activities. The accreditation certificate is the result of an internal assessment process that was executed by members interested in the university.

With this modelling scenario we show how current quality proposals do not integrally cover some relevant issues in MDE projects. This modelling scenario helps to identify the applicability of some of the quality works on MDE identified in Sections 2.4 and 2.5 and emerging quality issues (Section 3.4) as a consequence of using modelling languages in the development of an Information System.

A quality evaluation proposal that comes from one of the primary authors identified in Section 2.5 was used to analyze this modelling context (the *physics* of notations proposed in (Moody, 2009)). Even though the quality proposal meets its primary purposes in the analysis of the models and modelling languages involved, other quality issues emerge but they were not covered by the proposal. These issues influence the adoption of a model-driven initiative to manage concerns in information systems.

This information system is characterized by:

- The presence of multiple academic/administrative stakeholders from different areas of knowledge, participating collaboratively in the development of strategies for the generation/management of evidence according to the descriptive models of quality required, and the monitoring of the multiple sub-processes of quality instantiated in the university.
- The alignment with quality descriptive models that define the quality criteria. These include the self-evaluation guides issued by the National Accreditation Council (CNA)[15] under the Ministry of Education of Colombia, as well the ISO 9001 -2015 standard and the Colombian technical standard NTCGP 1000: 2009. The NTCGP 1000: 2009 is a management standard directed towards the evaluation of an institution' performance in terms of quality and social satisfaction during the delivery of services by government entities.
- The development of an organizational culture that is oriented towards the continuous improvement management of the university in the business processes. The support of this goal is the Integrated Management System[16], which is a web platform where the specification of processes, procedures, and associated institutional formats is published. The related application scenario is framed within the business process called self-assessment for the accreditation and re-accreditation of an undergraduate or graduate program.

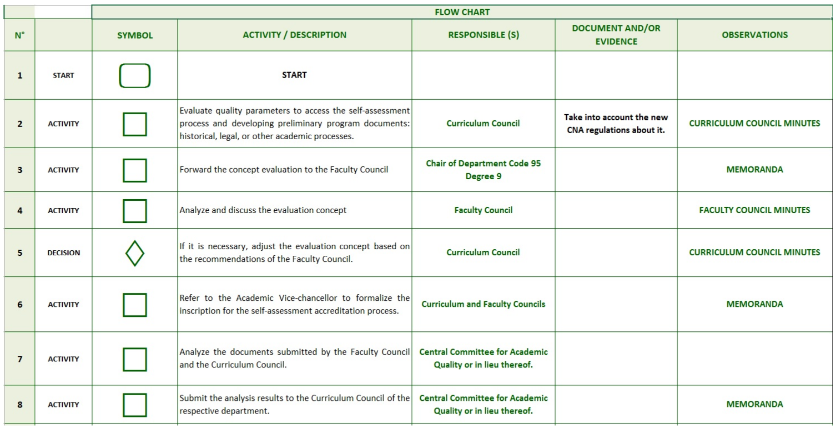
A strategy for the collaboration between academic experts and researchers in information systems was developed for the design, construction and deployment of the information system. Its purpose is to formulate conceptual, methodological, and technological tools that support the processes of accreditation and assurance of quality. Each group used modelling languages to represent the phenomena of interest. The panel of experts in quality specified a model for academic quality process[17] using a specific variation of the *Flowchart* diagram (a notation selected by those responsible for the integrated management system of the University Quindío to model the processes of the organization). The group of researchers in information systems employed the proper languages of software modelling and data to conceptually support the design and implementation of software platforms for different parts of the accreditation process. The use of different modelling languages for the process of design and construction of the academic quality system favors the process specification through the contributions of the parties involved (views). Three types of models were used in the conceptual modelling of the project:

- *Business process models*: This part of the application design focuses on the modelling of the processes undertaken at the University of Quindío, which are oriented towards business

[15] `http://www.cna.gov.co/1741/channel.html`

[16] Available at `http://www.uniquindio.edu.co/planeacion/publicaciones/sistema_integrado_de_gestion_1_pub`

[17] Available at `http://www.uniquindio.edu.co/planeacion/descargar.php?idFile=19777`



		FLOW CHART				
N°		SYMBOL	ACTIVITY / DESCRIPTION	RESPONSIBLE (S)	DOCUMENT AND/OR EVIDENCE	OBSERVATIONS
1	START	▢	START			
2	ACTIVITY	□	Evaluate quality parameters to access the self-assessment process and developing preliminary program documents: historical, legal, or other academic processes.	Curriculum Council	Take into account the new CNA regulations about it.	CURRICULUM COUNCIL MINUTES
3	ACTIVITY	□	Forward the concept evaluation to the Faculty Council	Chair of Department Code 95 Degree 9		MEMORANDA
4	ACTIVITY	□	Analyze and discuss the evaluation concept	Faculty Council		FACULTY COUNCIL MINUTES
5	DECISION	◇	If it is necessary, adjust the evaluation concept based on the recommendations of the Faculty Council.	Curriculum Council		CURRICULUM COUNCIL MINUTES
6	ACTIVITY	□	Refer to the Academic Vice-chancellor to formalize the inscription for the self-assessment accreditation process.	Curriculum and Faculty Councils		MEMORANDA
7	ACTIVITY	□	Analyze the documents submitted by the Faculty Council and the Curriculum Council.	Central Committee for Academic Quality or in lieu thereof.		
8	ACTIVITY	□	Submit the analysis results to the Curriculum Council of the respective department.	Central Committee for Academic Quality or in lieu thereof.		MEMORANDA

Fig. 10 Current self-assessment process diagram for the University of Quindío (partial view).

experts and the people who interact with the processes at the university. These models are intented for users of the processes that have no prior knowledge in order to facilitate the understanding of the processes.

- *Business and system models*: These models focus on the design and subsequent implementation of derived software applications to support the information system of academic quality, where everything that a software system needs to fulfill customer requirements must be specified. UML models are employed using class, sequence, use case, state, and components diagrams. In addition, a proposal of stereotyped UML formulated by RUP (Kruchten, 2000) is used to model business processes by applying the business modelling discipline defined in this methodological framework. Researchers with different profiles made these models: experts in accreditation and academic quality processes, experts in software engineering, senior/advanced software developers, and data experts. A model-based approach is used to produce the source code of the applications from the models made by the researchers.
- *Data Model*: These models cover the design of the database required for the academic quality system using the core business concepts identified in the domain model made in UML (the class diagram with the most representative concepts of the business according to the business modelling discipline of the RUP). This type of design depends on the expert in data or DBA (Database Administrator) because of the complexity that data modelling can have.

The complexity that is inherent in the development of the academic quality system and the parties involved is the rationale for using multiple modelling languages to help fulfill the interests of each role that is in charge of the implementation of the information system at the University of Quindío. These modelling languages include:

- Flowchart: the language used for making the process flow diagrams.
- UML: the language used for the analysis and design of software.
- E/R: Models used for verifying the design of the database.

A.1 Application of multiple models

Fig. 10 shows a partial view of the self-assessment process for *accreditation and re-accreditation purposes of an undergraduate or graduate program.* Fig. 11 presents the adaptation of the flow diagram notation used in the specification of business processes for the University of Quindío. This view corresponds to the participantion of the experts in the business information system and in the assurance of academic quality processes.

The modelling of business processes is done by using a notation that is particularly suited for experts in institution processes. This notation prioritizes simplicity and a small number

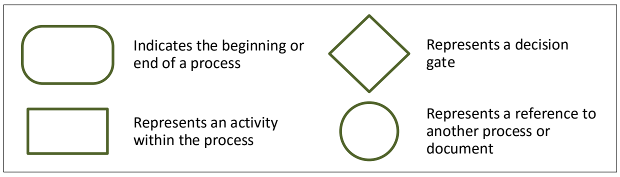

Indicates the beginning or end of a process
Represents a decision gate
Represents an activity within the process
Represents a reference to another process or document

Fig. 11 The conventions used for the flowchart adaptation at the University of Quindío.

of notational constructs to represent the process components accurately. None of the quality standards used for the implementation of quality policies (ISO 9001, NTC GP 1000, CNA) requires a specific graphic language; instead, these standards grant freedom for the modelling processes to be performed autonomously at the discretion of the organization.

Figures 12 to 16 present the conceptual models that are formulated by researchers and experts in information systems (mostly in UML) to address the various considerations associated with academic quality and the derived software platforms (publication of information related to academic quality processes, document management framed in quality contexts, document distribution of quality processes supports, and management of activities).

Due to the methodological alignment with RUP, a UML profile is used for business modelling. Then, the researchers formulate system models. The following models belong to the module of *Memoranda Management System within the Context of the Information System of Institutional Accreditation*[18].

A.1.1 Business modelling models

In order to understand the organization (i.e., detect current problems, identify improvement potential, identify users, workers, and parties, etc) several stereotyped UML models were employed following the RUP methodological framework (Figures 12 and 13). Fig. 12 -part A - shows the model of *business use cases.* This model illustrates the organization by management process areas of the university. Related business processes are identified as use cases (in light blue). For purposes of readability, they are grouped using standard UML packages. The business use case is a modelling of each business goal and its respective roles. It is used to identify the roles and different deliverables of the works performed.

The model of business use case also contains the business use cases realization (Fig. 12 -part B) as part of the business analysis model defined in RUP. A realization of a business use case describes how the workflow is in terms of the business objects and their collaboration. A diagram of activities and a diagram of business objects are defined in the realization of a business use case.

The business process model (Fig. 12 - part C) is a set of logically related tasks that are carried out to generate products and services. A stereotyped UML activity diagram represents this model, where the business entities that are involved in the process tasks are also identified.

The business modelling discipline of RUP considers all the *things* or *something of value* that are observable during the performing of business processes. For this, researchers used the models shown in Fig. 13. The business entity model (Fig. 13 - Part A) represents an important part of the information that is handled by business actors and business workers. The business object model (Fig. 13 - Part B) shows the relationship between the business entities associated with different business use cases and the workers associated to those cases. The model serves to show the limits of the business process considered in each business use case.

Finally, a state machine model is used to define the life cycle of the information entities at the University of Quindío. Each state considers a set of specific software features to manage the state associated with an entity at any time during the execution of the process. Fig. 13 - part C - shows a sample lifecycle for a communication in the context of academic quality.

[18] Figures 12 to 17 show the current application of models in the *Information System of Institutional Accreditation* project. For this reason, these diagrams are presented as they are currently in use. Copyright©SINFOCI Research Group, University of Quindío, 2015.

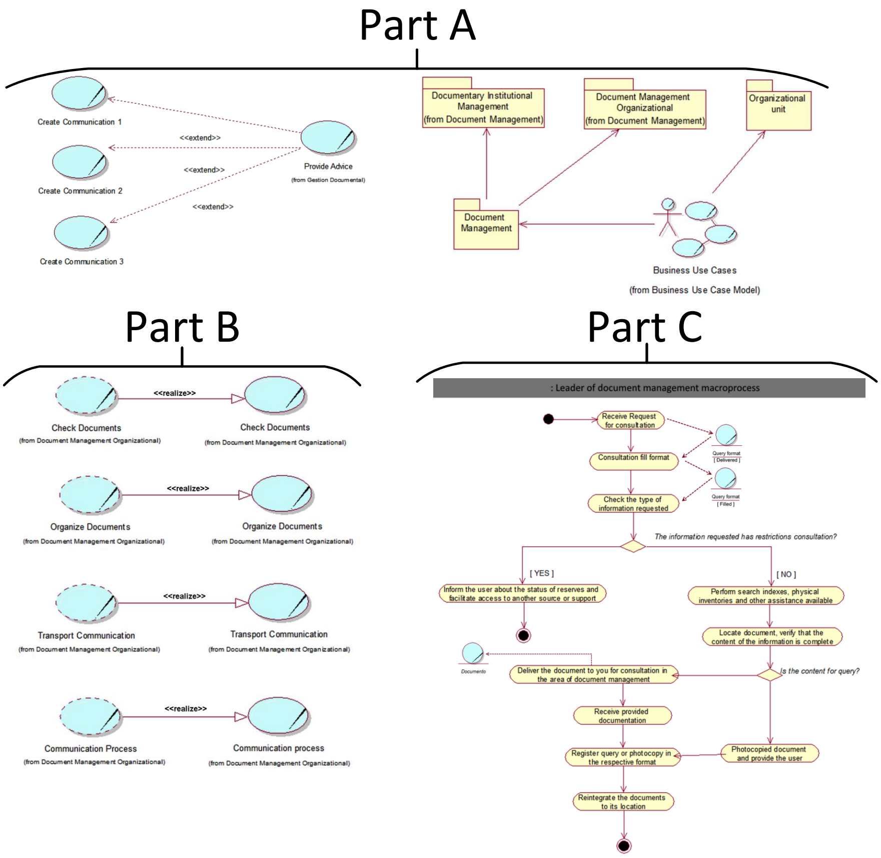

Part A
Create Communication 1
Create Communication 2
Create Communication 3
<<extend>>
<<extend>>
<<extend>>
Provide Advice
(from Gestion Documental)
Documentary Institutional Management
(from Document Management)
Document Management Organizational
(from Document Management)
Organizational unit
Document Management
Business Use Cases
(from Business Use Case Model)
Part B
Check Documents
(from Document Management Organizational)
<<realize>>
Check Documents
(from Document Management Organizational)
Organize Documents
(from Document Management Organizational)
<<realize>>
Organize Documents
(from Document Management Organizational)
Transport Communication
(from Document Management Organizational)
<<realize>>
Transport Communication
(from Document Management Organizational)
Communication Process
(from Document Management Organizational)
<<realize>>
Communication process
(from Document Management Organizational)
Part C
: Leader of document management macroprocess
Receive Request for consultation
Query format [Delivered]
Consultation fill format
Query format [Filled]
Check the type of information requested
The information requested has restrictions consultation?
[YES]
[NO]
Inform the user about the status of reserves and facilitate access to another source or support
Perform search indexes, physical inventories and other assistance available
Locate document, verify that the content of the information is complete
Documento
Deliver the document to you for consultation in the area of document management
Is the content for query?
Receive provided documentation
Register query or photocopy in the respective format
Photocopied document and provide the user
Reintegrate the documents to its location

Fig. 12 Business modelling models (I)

A.1.2 System models

Once the definition of business processes has been completed, use cases are derived at the software system level by a relationship of traceability whose origin is found in automatable activities of the business process analyzed.

Fig. 14 - part A - partially shows the features that are implemented for the module of *memoranda management* software of the information system for academic quality. Models of system classes (Fig. 14 - part B) generate the associated source code (logical view of the application) and sequence diagrams (functional allocation of responsibilities among objects) of Fig. 14 - Part C. These diagrams (along with their associated specification) are delivered to the project developers who generate the source code in the platforms and development environments that are defined by the technical experts.

Other non-UML systems models were used to conceive and manage specific system views of the *Information System of Institutional Accreditation*. Fig. 15 shows the *Data model* in the E/R notation. Due to the relational support used in the technological implementation of the modules associated with the quality system, a conceptual representation of the entities associated to the domain addressed by each module is made. This conceptual representation defines the semantics associated with the entities, the consistency constraints at the data level in order to preserve the integrity of the module once it deploys organizationally.

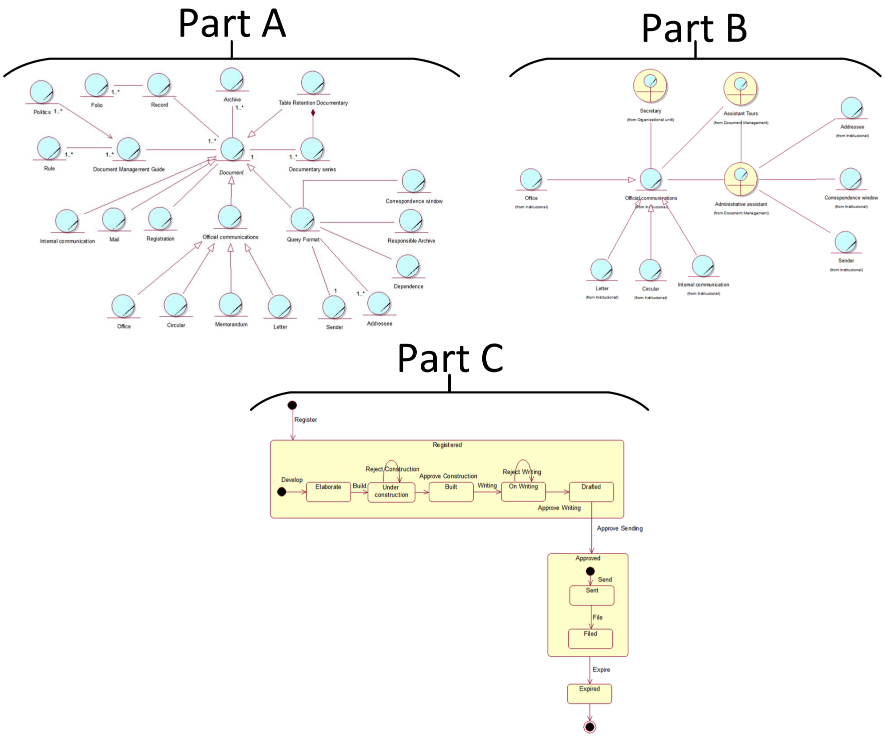


Fig. 13 Business modelling models (II)

Additionally, as part of the process of architectural decision-making for developing software modules, models elaborated in informal notations are used to address problems associated with specific quality attributes and to facilitate the identification of architectural tactics in the management of these attributes. Fig. 16 shows an example of a diagram that was developed to discuss the aspects of global integration and the consistency of the information system (taking into account the presence of multiple software modules). The aim of these diagrams is to facilitate the description of architectural alternatives in the consultation and judgment processes so that the consequences and impact of each architectural strategy formulated are easily addressed.

Finally, Fig. 17 depicts the software products obtained from the conceptual models identified by the researchers to support specific elements of the academic quality system.

A.2 The first signs of quality problems

The first signs of quality problems associated with the use of multiple models and different modelling languages can be observed. The first problems can be found by analyzing the visual language used by experts and organizational stakeholders to represent the business processes of the university, since it is the self-assessment process for accreditation and re-accreditation of an undergraduate or graduate program.

The researchers decided to evaluate the graphical notation using the theory of Physics of Notations (PoN) by D.L. Moody (Moody, 2009), which is the most frequently published. The application of this theory provides a scientific basis for comparison, evaluation, improvement, and construction of visual notations used in an organization. The PoN theory proposes nine

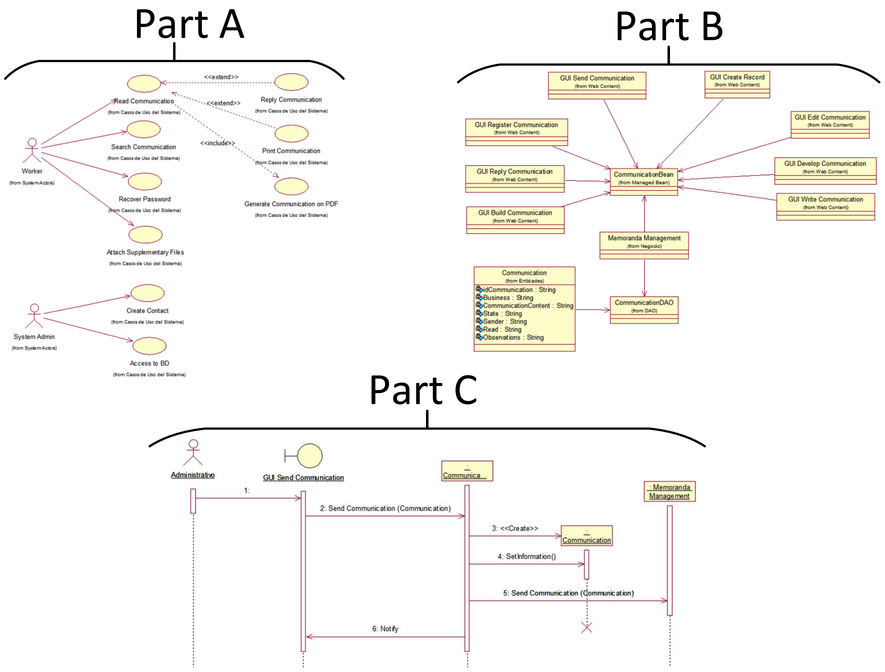


Fig. 14 System models (I)

principles that can be successfully used to assess visual languages of graphic modelling (*Cognitive Integration*, *Cognitive Fit*, *Manageable Complexity*, *Perceptual Discriminability*, *Semiotic Clarity*, *Dual Coding*, *Graphic Economy*, *Visual Expressiveness*, and *Semantic Transparency*).

The institution does not use a standard visual language for modelling its business processes. The variant of the flow chart used by the university in the modelling of its processes does not preserve the semantics that is used for this type of notation, which causes the process model to be unclear for the roles that interact with them. Thus, the application of PoN helps validate the flowchart version created in the institution by applying the principles that this theory proposes.

This type of graphic language is not suitable for modelling business processes or complex systems because of its simplicity. In these cases, it is possible to find many other languages that are also appropriate such as BPMN or UML activity diagram. However due to the lack of knowledge about different alternatives for process modelling, the migration of these processes to other languages has not been done.

The application of the PoN principles in the flowchart diagram variant used in process modelling at the University of Quindío is presented in the following sections.

A.2.1 Semiotic clarity

This principle establishes a one-to-one correspondence between the semantic constructions and the graphic symbols of visual language. When there is not a one-to-one correspondence between the analyzed symbols and their respective semantics, at least one quality problem generated in the notation which is related to *Symbol Deficit*, *Symbol Redundancy*, *Symbol Overload*, or *Symbol Excess*.

Fig. 18 shows the analysis of notational elements employed in the variant flowchart applied at the University of Quindío compared to the original semantic constructs from the flowchart.

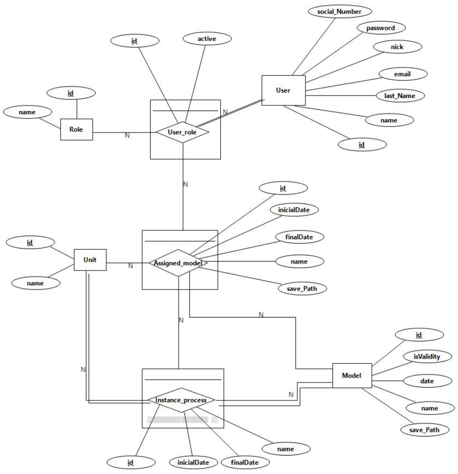


Fig. 15 System data model (partial view).

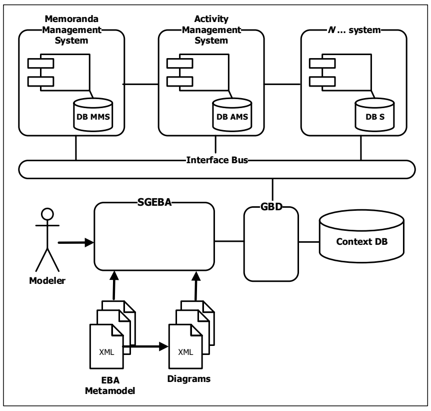


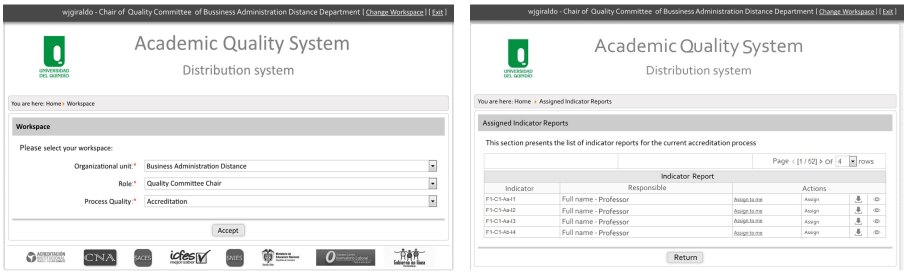
Fig. 16 Diagram example for the rationale of an architecture decision.

The simplicity that is applied at the University of Quindío for conducting the flowcharts is shown in this analysis because not all the symbols originally formulated by the notation are used. As a result, out of the 16 original notation symbols contained in the university flowchart, only 3 symbols that have the same semantic construct and another construct with a different meaning are used. This analysis found two specific anomalies regarding the principle of semiotic quality, *Symbol Deficit* and *Symbol Excess.*

The *Symbol Deficit* anomaly found represents the lack of 13 symbols by the university in order to meet the standards of a flowchart. For the *Symbol Excess* problem, the use of the visual element **internal connector** is contrasted (Fig. 19), identifying the meaning given in the description of processes of the University of Quindío and its original semantics according to specifications of the flowcharts (ISO, 1985).

A.2.2 Perceptual Discriminability

This principle is related to the ease and perception with which the symbols used in a graphical notation can be distinguished from each other. Although this principle is supported by the



wjgiraldo - Chair of Quality Committee of Bussiness Administration Distance Department [Change Workspace] [Exit]

Academic Quality System

Distribution system

You are here: Home ▸ Workspace

Workspace

Please select your workspace:

Organizational unit:* Business Administration Distance

Role:* Quality Committee Chair

Process Quality:* Accreditation

Accept

wjgiraldo - Chair of Quality Committee of Bussiness Administration Distance Department [Change Workspace] [Exit]

Academic Quality System

Distribution system

You are here: Home ▸ Assigned Indicator Reports

Assigned Indicator Reports

This section presents the list of indicator reports for the current accreditation process

Page ‹ [1 / 52] › Of 4 rows

Indicator Report

Indicator	Responsible	Actions			
F1-C1-Aa-I1	Full name - Professor	Assign to me	Assign	⬇	👁
F1-C1-Aa-I2	Full name - Professor	Assign to me	Assign	⬇	👁
F1-C1-Aa-I3	Full name - Professor	Assign to me	Assign	⬇	👁
F1-C1-Ab-I4	Full name - Professor	Assign to me	Assign	⬇	👁

Return

Fig. 17 Examples of software products obtained from conceptual models.

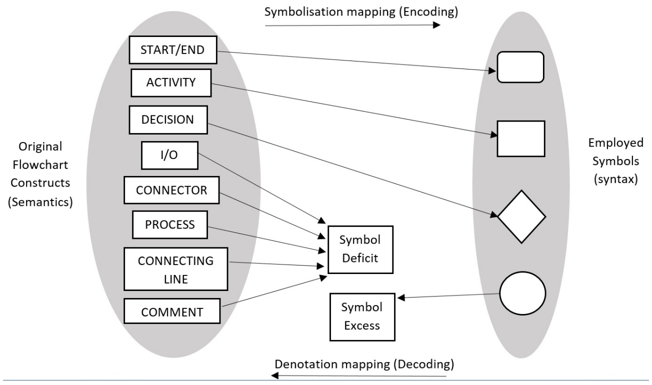

Symbolisation mapping (Encoding)
Original Flowchart Constructs (Semantics)
START/END
ACTIVITY
DECISION
I/O
CONNECTOR
PROCESS
CONNECTING LINE
COMMENT
Symbol Deficit
Symbol Excess
Employed Symbols (syntax)
Denotation mapping (Decoding)

Fig. 18 Principle of Semiotic Clarity: there should be a 1:1 correspondence between semantic constructs and graphical symbols

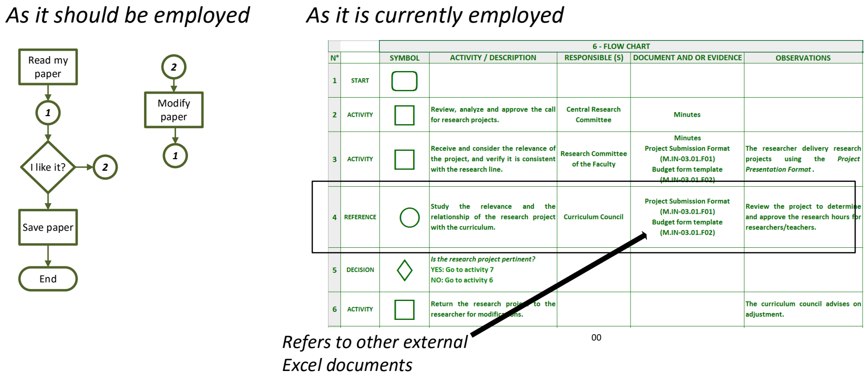
specific adaptation of the flowchart conducted at the university, the main problem found in the analysis of perception is simplicity due to the number of symbols used. This can be seen as something that is relatively handy when making model interpretation of the business process. However, given the complexity of a business process of an organization, it is not feasible to conduct a modelling with so few symbols, since it loses too much of the useful information that provides a better understanding and proper execution of the process.

A.2.3 Semantic Transparency

The principle of semantic transparency refers to the ease of identification of the semantic meaning of a symbol that is used in a graphical notation. This principle considers four possible classifications for the analyzed symbols of the visual language:

- *Semantically Perverse*: When the symbol is observed, it is not easy to identify its meaning.
- *Semantically Opaque*: When the symbol is observed the person arbitrarily relates it to something known in order to identify its meaning.
- *Semantically Translucent*: In order to know the meaning of the symbol, the person requires prior explanation.
- *Semantically Immediate*: The meaning of the analyzed symbol can be identified easily without prior explanation.

The notation used for the modelling of processes at the University of Quindío identifies two semantically transparent symbols (Fig. 20 - left) since they preserve the semantic construct of the flowcharts. Thanks to this, it is easy to identify their meaning (*semantically immediate*). However the presence of the *semantically opaque* category is also evident (Fig. 20 - right) because the users of the business process (when noting some of the symbols by intuition and



		6 - FLOW CHART				
N°		SYMBOL	ACTIVITY / DESCRIPTION	RESPONSIBLE (S)	DOCUMENT AND OR EVIDENCE	OBSERVATIONS
1	START	▢				
2	ACTIVITY	□	Review, analyze and approve the call for research projects.	Central Research Committee	Minutes	
3	ACTIVITY	□	Receive and consider the relevance of the project, and verify it is consistent with the research line.	Research Committee of the Faculty	Minutes Project Submission Format (M.IN-03.01.F01) Budget form template (M.IN-03.01.F02)	The researcher delivery research projects using the *Project Presentation Format* .
4	REFERENCE	○	Study the relevance and the relationship of the research project with the curriculum.	Curriculum Council	Project Submission Format (M.IN-03.01.F01) Budget form template (M.IN-03.01.F02)	Review the project to determine and approve the research hours for researchers/teachers.
5	DECISION	◇	*Is the research project pertinent?* YES: Go to activity 7 NO: Go to activity 6			
6	ACTIVITY	□	Return the research project to the researcher for modifications.			The curriculum council advises on adjustment.

Fig. 19 Comparison between the symbols used at the University of Quindío and the symbols used in the semantic construct of the flowcharts

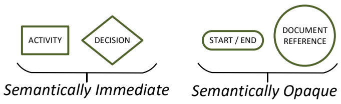


Fig. 20 Symbols used in the university that meet the *semantically immediate* / *semantically opaque* categories.

perception) relate what they observe to any known symbol. This gives a meaning that is not generally correct. At the University of Quindío there are symbols for start/end, and there is another symbol for refering to documents or processes.

A.2.4 Visual Expressiveness

The principle of expression evaluates the number of visual variables used and the range of values (capacity) of these variables. It considers the use of space of graphic design and the variation in the whole visual vocabulary. Table 10 presents the identified values for the variables associated with this principle for the language used in the modelling processes of the University.

Visual Variable	Usage
Shape	The visual language uses default figures in Excel (simple figures associated with 2D Flowchart Diagrams).
Brightness	The activities are represented by squares in bold. The events and decisions are represented by a box with rounded corners and diamonds, respectively.
Spatial Location (x,y)	Each symbol is located in each cell of the table generated in Excel. Guidance does not use arrows or lines.
Size	The symbols have a predetermined size that cannot be modified.
Colour	Use of symbols in white with green edge.
Texture	Not Used
Orientation	Not Used

Table 10 Visual variables of the flowchart notation used at the University of Quindio

The main abnormality is the lack of guidance (arrows, lines, or useful symbols) to denote the process flow of the diagram, which restricts the browsing in the business process modeled.

This reduces the diagram to a top-down sequential specification. The sharp demarcation in the application of colors creates identification problems for parts of the process, which affects its cognitive assimilation.

A.2.5 Complexity Management

This principle evaluates the ability of visual languages to present large amounts of data without overloading the human mind. This principle refers to schematic complexity, which is based on the number of elements (instances or symbols) used in the diagrams. When analyzing this principle on the models of the business processes of the university, a high level of complexity due to the high number of activities (see Fig. 10) is presented. This hinders the understanding and implementation of the process. To reduce the levels of schematic complexity in models of business processes, subprocesses are generally used to group activities. This minimizes the number of symbols used in the modelling of the process and achieves a better understanding of the workflow.

A.2.6 Dual Coding

This principle measures the use of text and graphics that are used together to transmit information. Specifically, the use of labels (text) plays a critical role in the interpretation of business diagrams since it defines and clarifies the semantics of the processes directly on the diagrams (i.e., the correspondence with the real-world domain).

The symbols used in the modelling of business processes at the University of Quindío have text labels to help interpret the flowcharts. The graphics used are inside Excel cells, which have several adjoining cells with associated text that provide information for the people who interact with these diagrams. The main drawback of these diagrams used is their excessive emphasis on the textual representation (Fig. 10). The visual elements fulfill a decorative function instead of a reasoning and communication function about the business process itself. The interpretation and expressiveness of the process models are directly affected by the excessive simplicity of the notation. The text itself becomes the central element of each diagram.

A.2.7 Graphic Economy

This principle states that the graphical complexity of a notation must be cognitively manageable. The number of visually distinct symbols of the notation indicates the complexity of a chart. This principle is critical to help the understanding and expressiveness of process models. The graphical notation used at the University of Quindío is too minimalist (there are only 4 symbols out of the 16 originally specified in the flowcharts). This makes it less useful for the modelling of systems or complex processes given their lack of semantic support from the specific syntax employed.

A preliminary application of the PoN method identifies the shortcomings of the modelling language that is currently used at the University of Quindío. This application highlighting its simplicity for the specification of the process models since the flowcharts do not meet the requirements for the modelling of processes and complex systems.

A.3 Limitations of the selected approach to evaluate the quality of the models of the modelling scenario

The processes for the management of academic quality are highly changing and dynamic, mainly because of regulatory updates from the authorities that govern academic quality in Colombia (the Ministry of Education and CNA). These changes affect organizations that voluntarily apply for accreditation processes, as is the case of the University of Quindío. Additionally, there are specific organizational conditions (administrative restructuring, updating of procedures, involvement of experts from different areas of knowledge, etc.) within the institution that affect the quality process models, which in turn affect the models that conceptually support the information systems generated.

The office of Planning and Development of the University of Quindío starts the exploration of a strategy of business process management using the BPM discipline with its associated notation (BPMN). To do this, in conjunction with the researchers involved in the project, a systemic approach for the selection of BPM tools (commonly known as BPM Suites) applied. This assessment was reported in (Gallego et al., 2015). Once the most suitable BPM Suite for the institution was selected, the researchers formulate an initial proposal in BPMN for the business process of self-evaluation from the specification presented in Fig. 10. A model containing 14 roles, 67 activities, and 67 attachments was obtained.

The proposed model was presented to them. Both the experts and the people from the planning department had difficulty understanding the model due to the high cognitive load and information present in the diagram generated. As reported in (Gallego et al., 2015), the researchers formulated an intervention to the original specification of the model to facilitate understanding by the business experts. This clearly shows the emergence of quality issues such as expressiveness, understandability, completeness, and appropriateness of the models.

From the perspective of researchers in information systems, the system models in UML and other languages (with their conceptual support) contribute to the creation of communication scenarios and documentation on which they make decisions that are related to a specific technological implementation. The modelling tools that are used support the automatic generation of source code (MDD). However, the emphasis on conceptual modelling of the different components of the information system require an extra effort for their subsequent translation into a specific platform of implementation. This is due to the particularities that must be developed in order to support the essential features of any model that formulated in the project on that platform.

Despite the considerable number of system conceptual models generated by the research group (especially the use of the UML profile for business modelling), their importance was perceived with relative apathy by the business experts at the University of Quindío. This was mainly due to the lack of alignment between the models of the information system and the specification of the models of organizational processes. Although the generation of information system platforms was delegated to the researchers because of the innovative nature of the conceptual models used to develop an information system for academic quality, the system models are limited exclusively to the use of roles for analysts, designers, and software developers. Therefore, in order to avoid suspicion and loss of confidence in the system models by the business experts and the users of the self-assessment process, the development team had to generate incremental versions of the components of the information system modelling. This produced software solutions that allowed the users and people involved in the self-evaluation process to appreciate the feasibility of the innovative proposals made by the researchers. In this case, the models contained reference information to support implementation decisions, but they were not used to automatically generate the underlying infrastructure of code (a model-based approach instead of model-driven one was used).

While models in this project played a strategic role at the organizational and conceptual support level of an information system with computational implementation, there was a decoupling between the organizational modelling and the system modelling. This caused duplication of the modelling effort and lack of mechanisms for traceability that covered the evolution of business aspects for their respective technological implementation.

In the implementation at the University of Quindío the understandability of models was important, but there are still other questions that remain open. For example, the suitability of UML models to address organizational concerns are not covered by current modelling efforts at the University of Quindío.