Software Quality Journal (2021) 29:275–307
https://doi.org/10.1007/s11219-021-09554-1

Empirical validation of a quality framework for evaluating modelling languages in MDE environments

Fáber D. Giraldo[1,3] · **Ángela J. Chicaiza**[1] · **Sergio España**[2] · **Óscar Pastor**[3]

Accepted: 15 March 2021 / Published online: 3 April 2021

Abstract

In previous research, we proposed the *multiple modelling quality evaluation framework* (MMQEF), which is a method and tool for evaluating modelling languages in model-driven engineering (MDE) environments. Rather than being exclusive, MMQEF attempts to complement other methods of evaluation of quality such as SEQUAL. However, to date, MMQEF has not been validated beyond some concept proofs. This paper evaluates the applicability of the MMQEF method in comparison with other existing methods. We performed an evaluation in which the subjects had to detect quality issues in modelling languages. A group of expert professionals and two experimental objects (i.e. two combinations of different modelling languages based on real industrial practices) were used. To analyse the results, we applied quantitative approaches, i.e. statistical tests on the results of the performance measures and the perception of subjects. We ran four replications of the experiment in Colombia between 2016 and 2019, with a total of 50 professionals. The results of the quantitative analysis show a low performance for all of the methods, but a positive perception of MMQEF.*Conclusions:* The application of modelling language quality evaluation methods within MDE settings is indeed tricky, and subjects did not succeed in identifying all quality problems. This experiment paves the way for additional investigation on the trade-offs between the methods and potential situational guidelines (i.e. circumstances under which each method is convenient). We encourage further inquiries on industrial applications to incrementally improve the method and tailor it to the needs of professionals working in real industrial environments.

Keywords Quality · Model-driven engineering · Quality frameworks · Empirical evaluation · The MMQEF method

1 Introduction

The model-driven engineering (MDE) paradigm is a subfield of software engineering that focuses on *models* as the main artifacts of an engineering process instead of source code in a traditional programming language. In MDE, source code is a kind of model

✉ Ángela J. Chicaiza
ajchicaizao@uqvirtual.edu.co

Extended author information available on the last page of the article or it can be generated from other models. Models are defined by *modelling languages*, which provide key elements for specifying models such as *abstract syntax* (i.e. domain concepts), *concrete syntax* (notation), *semantics*, and *pragmatics* or guides about the appropriate use of language. Due to the current facilities and technical environments for defining languages, one of the main trends in the modelling field is *domain-specific languages* (DSL), which allows models that are closer to an application domain to be generated (Bézivin, 2005; Mernik et al., 2005; da Silva, 2015).

Since *models* are the main artifact in MDE methods, the *quality of the modelling languages* is important for MDE environments. This feature is particularly important considering the current applications of models and modelling languages, such as the case reported in (Wortmann et al., 2019). There are methods to evaluate the quality of modelling languages, but when they are applied in MDE environments, some problems appear due to the features of these environments. Since multiple modelling languages are used together, model transformations add a level of complexity to the evaluation of quality because the suitability of the source and target languages must be ensured, and the code generation requires quality requirements of modelling languages and models (e.g. the construct deficit in one language or incomplete models may have an effect on the resulting source code).

There is great diversity among researchers about the definition of quality in MDE contexts (Giraldo et al., May 2014). Considering that *quality* is not a concrete manifestation (or physical component) of modelling artifacts, there are many facets or levels of quality. The different meanings of quality have not been acknowledged, which is an open problem. Some of the identified quality definitions have associated frameworks for evaluating quality issues. These frameworks present specific validation procedures for demonstrating their applicability and utility in accordance with the given quality definitions.

Since 2013, we have performed a systematic review in order to identify the main trends in the definition of quality in MDE contexts (Giraldo et al., 2018a). To date, there are twenty-nine works that provide an explicit definition of quality; of these, only nine works have an associated validation procedure (Challenger et al., 2015; Espinilla et al., 2011; Grobshtein & Dori, 2011; Hindawi et al., 2009; Lange & Chaudron, 2005; Le Pallec & Dupu Chessa, 2013; Lopez-Fernandez et al., 2014; Maes & Poels, 2007; Merilinna, 2005). When we used the classification of evaluation techniques that were previously proposed in (Siau & Rossi, 1998) on the nine identified works, we found five works that reported laboratory experiments, two works that used metrics, and other individual works that used survey, metamodelling, and case studies. The laboratory experiments use different evaluation procedures with participants, surveys, and implementation of software tools to demonstrate the applicability of the work that is proposed. The only common aspect in these experiments is their specificity with regard to the scope of each quality definition.

This paper presents the design and results of a controlled experiment that was performed between 2016 and 2019 to evaluate the applicability and use of approaches for evaluating quality issues in modelling languages in MDE projects. In accordance with (Siau & Rossi, 1998), which defines some approaches for evaluating information modelling methods, this experiment can be considered to be an *empirical evaluation technique*. The design of the experiment was done in accordance with the guidelines described in (Wohlin et al., 2012) for experimentation in software engineering scenarios. We use the guidelines of (Jedlitschka et al., 2008) for reporting the execution and results of the empirical validation. Therefore, the remainder of this article is structured as follows: Section 2 presents the context of the experiment. Section 3 presents the planning of the performed experiment. Section 4 presents the results obtained from the experiments with their corresponding analysis. Section 5 presents a discussion about the findings and results that were obtained. Finally, the conclusions are presented.

2 Background

2.1 Overview of quality methods for MDE

Modelling languages are the main artifacts of MDE environments. Modelling languages are conceptual tools that allow viewpoints and concerns to be conceived and addressed. Modelling languages also allow representations (views) of these concerns to be generated. The evaluation of quality in modelling languages is a critical task considering their key role in a model-driven project.

Due to the multiple (and even ambiguous) interpretations of the model-driven paradigm, there is not a clear conception of *quality* as the fulfillment of some modelling artifact with its associated specification. With regard to quality evaluation procedures, the literature in the model-driven engineering field provides examples of the applicability of quality evaluation frameworks in the following: over specific modelling initiatives (e.g. the *Physics of Notations* - PoN (Moody, 2009) in Big Data analytics (Khalajzadeh et al., 2020) and multiagent (Miranda et al., 2019) modelling languages); specific quality evaluation approaches in specific modelling domains (e.g. the embedded systems domain (Arslan & Kardas, 2020), the multiagent domain (Asici et al., 2021; Alaca et al., 2021)); comparison between modelling initiatives (Santos et al., 2020); and analysis of modelling approaches from the perspective of potential users of the modelling initiative (Shin, 2019). In order to tame the heterogeneity and multiplicity of related approaches, the authors in (Fischer & Strecker, 2018) compile several works that report evaluation procedures in conceptual models and modelling languages.

Some proposals for evaluation of quality in MDE have different underlying theories (e.g. semiotic, cognitive process, linguistics, psychological, and others). In addition, the application of the previous proposals covers different artifacts of a model-driven project (i.e. concrete syntax, abstract syntax, and other specific properties of modelling languages). The artifacts under evaluation are defined within the scope of the quality method. Therefore, we can find methods with a broad scope (e.g. the SEQUAL framework (Krogstie, 2012)), as well as other methods with a specific application (e.g. PoN).

Previous quality methods for MDE have been formulated for addressing quality issues in modelling languages. The Physics of Notations - *PoN* defines nine principles for evaluating the perceptual properties of visual modelling languages in order to improve their understandability for the final users of the languages. These are: *Semiotic Clarity, Perceptual Discriminability, Semantic Transparency, Complexity Management, Cognitive Integration, Visual Expressiveness, Dual Coding, Graphic Economy*, and *Cognitive Fit*. The evaluation is performed by interpreting these principles and applying them to the language to determine the fulfillment of the language with each one of them. In this way, the quality of the language is determined by whether or not the language meets each principle and how the language must be fixed in order to be compliant with a conflictive principle. The *6C Goals* framework (6C) (Mohagheghi, 2009) is a set of desirable properties or quality goals (*Correctness, Changeability, Consistency, Comprehensibility by humans, Confinement,*, and *Completeness*) that were initially deduced for models. The framework defines each goal so that the analyst is responsible for determining whether or not a model meets the goals.

The SEQUAL framework provides a wide conceptual foundation for quality in model-driven and model-based initiatives that comes from semiotic theories. SEQUAL defines the *quality of models* (Krogstie, 2012a) separately based on seven semiotic levels (*Physical, Empirical, Syntactic, Semantic and Perceived Semantic, Pragmatic, Social, and Deontic*).

The framework also defines the *quality of modelling languages* (Krogstie, 2012b) that is based on six quality categories (*Domain appropriateness, Comprehensibility appropriateness, Participant appropriateness, Modeller appropriateness, Tool appropriateness, and Organizational appropriateness*).

The multiple modelling quality evaluation framework method (MMQEF) (Giraldo et al., 2018b; Giraldo et al., 2019) is a conceptual, methodological, and technological framework for evaluating quality issues in modelling languages and modelling elements by the application of a taxonomic analysis. It derives analytic procedures that support the detection of quality issues in modelling languages, such as the suitability of modelling languages, traces between abstraction levels, specification for model transformations, and integration between modelling proposals. MMQEF also suggests metrics to perform analytic procedures based on the classification obtained for the modelling languages.

MMQEF uses a taxonomy that is extracted from the Zachman framework for Information Systems (Zachman, 1987; Sowa & Zachman, 1992), which proposed a visual language to classify elements that are part of an IS. These elements can be from organizational to technical artifacts. The visual language contains a bi-dimensional matrix for classifying IS elements (generally expressed as models) and a set of seven rules to perform the classification. In this way, MMQEF defines the quality of a modelling language as the degree of its fulfillment with essential principles of IS that are defined in the Zachman reference architecture for IS. As an evaluation method, MMQEF defines activities in order to derive quality analytics based on the classification applied to modelling languages. The Zachman framework was chosen because it was one of the first and most precise proposals for a reference architecture for IS, which is recognized by important standards such as the ISO 42010 (ISO, 2011).

2.2 Problem statement & research objective

The validation of quality methods for MDE contexts, such as the 6C Goals, PoN, and SEQUAL, has challenges. To detect validation procedures for these frameworks, specific publications of the authors about the applicability of the frameworks must be accessed separately. Examples of validation procedures can be found in (Heggset et al., 2015). Some approaches that have been employed in validation are experiments, surveys, interviews, and questionnaires.

Besides the specific modelling scenarios that are required to demonstrate the application of the quality methods for MDE, the quality methods are not directly comparable with each other. Although it is true that the quality frameworks could be similar theories for Information Systems (Gregor, 2006), they differ in their purposes, scope, and procedures for the identification of quality issues. Reported validations present individual applications of the quality frameworks.

Using the template defined in (Wohlin et al., 2012b) for the definition of goals in experimentation processes, the main purpose of this experimentation is described as follows:

Analyse the MMQEF method

for the purpose of characterizing it

with respect to its applicability for finding quality issues for modelling languages

from the point of view of the researcher

in the context of professional experts analysing a scenario for the application of multiple modelling languages.

3 Design of the experiment

The experiment was formulated to identify the degree of applicability of the MMQEF method for evaluating quality in MDE contexts, specifically modelling languages as the main artifact of this paradigm. This evaluation also considers other quality methods that are formulated in the MDE literature so that MMQEF could be applied in similar conditions of practice, taking into account that the population of the experiment had no previous experience with quality methods for MDE.

A group of participants in Colombia (Spanish-speaking participants) applied the MMQEF method in a model-driven scenario that we had defined beforehand. During the experiment, the participants were asked to find quality evidence for modelling languages that are jointly applied to model an IS project. To do this, the participants used MMQEF and another quality framework for MDE which was freely chosen by each one of them.

Prior to the experiment, we identified some quality issues. We took advantage of a *post-mortem* analysis that was performed by the researchers who led the project. These researchers were given roles such as domain expert, modelling-data leader, and software engineering leader. Table 1 summarizes some of the quality issues for the experiment that were expected to be found. This list is not exclusive (i.e. other issues could be reported by the participants).

3.1 Experimental units

We used a *convenience sampling* approach to select the participants, who were contacted and invited based on their homogeneous knowledge and condition (academic or professional) for applying quality frameworks.

For the experiment, there was a total of fifty participants (Master's students and professionals) with previous knowledge of MDE and model-driven environments such as Eclipse Modelling Framework (EMF/GMF)[1] and JetBrains MPS[2]. The participants came from several software development companies; they had experience in roles such as senior software developers / software architects (33), software project managers (10), and teachers (7). They have expertise in software development projects and are currently working in software development companies, with an average of 5.7 years of work experience. Most of the participants were post-graduate students who are involved in a Software Engineering Master's program. The students were in a Master's course about *domain-specific language (DSL) design and implementation*. In addition, those students had previously taken two courses in MDE (Introduction to MDE and Applied MDE). Both (professionals and Master'students) were contacted by email and they voluntarily accepted to participate in the experiment.

For the design of the experiment, we used a *Probability-Paired comparison design* to avoid the influence of the quality evaluation methods in MDE during their application by the participants. For the experiments, the *paired comparison* was defined as presented in Table 2. For their participation, those invited were rewarded with free seminars and lunches/dinners.

---

[1] https://www.eclipse.org/modeling/emf/

[2] https://www.jetbrains.com/mps/

Table 1 Example of quality issues expected to be reported in the experiment

ID	Quality Issue	Derived from
QIPR01	There was no traceability from models to code.	MMQEF
QIPR02	There was no automatic code generation.	MMQEF
QIPR03	Decoupling between organizational modelling and system modelling.	MMQEF
QIPR04	Misalignment between the modelling languages used and the purposes of modelling.	MMQEF
QIPR05	Excessive stereotyping of the UML modelling language.	MMQEF
QIPR06	Excessive adaptation of languages to model business concerns.	MMQEF, SEQUAL
QIPR07	Lack of suitability analysis of modelling languages.	MMQEF
QIPR08	Lack of coverage of modelling languages. Some IS issues were not covered by modelling languages (e.g. data, interaction, architectural decisions).	MMQEF
QIPR09	There was no distinction of the purpose of the resulting models (i.e. there were models to communicate ideas, to automate the process, to make systems), but these purposes were not explicitly addressed.	MMQEF, SEQUAL
QIPR10	There was no integration between modelling languages.	MMQEF
QIPR11	Poor support for the deontic level (for organizational and system purposes).	MMQEF, SEQUAL
QIPR12	Lack of adequate tool support.	MMQEF
QIPR13	Lack of expressiveness of the modelling languages.	6C, MMQEF, SEQUAL
QIPR14	Lack of communication abilities for the performed modelling.	MMQEF

Prior to the use of quality methods, the participants were asked about their previous knowledge about key terms for MMQEF, MDE, modelling languages, and DSLs in order to determine whether or not their previous knowledge might eventually affect the application of the method. The set of terms used by MMQEF is not limited just to the application of the method. These terms are common concepts in the MDE terminology. Thus, the familiarity with these terms facilitates the performance of activities for evaluation of quality that are proposed in MMQEF. Previous knowledge and use of modelling languages and DSLs could also induce key MDE terms.

3.2 Experimental material

The objects that were used in the experiment were the following:

- The slides of the seminar about *quality in MDE (modelling languages)*.
- A summary of the MMQEF with the main blocks (components) of the method.

Table 2 Paired-comparison design for the validation of MMQEF

i - participant	Treatment 1	Treatment 2
Odd participant	MMQEF	Other method
Even participant	Other method	MMQEF

- A description about a modelling scenario where multiple modelling languages (≥ 2) are required for addressing specific IS concerns. This scenario was previously reported in (Giraldo et al., 2018a).
- A questionnaire for characterizing each participant.
- A questionnaire for reporting the quality issues that are identified by the participants (for both the alternative approach and the MMQEF treatment).
- A survey about the *Perceived ease of use* (PEU), *Perceived usefulness* (PU), and *Intention to use* (IU) for MMQEF. This survey was made using twelve statements with answers on a Likert scale of 1 to 5 (1: Strongly disagree, 2: Disagree, 3: Neither agree nor disagree, 4: Agree, 5: Strongly agree). In the Likert statements, we used explicit leading phrases about the applicability of MMQEF in order to more easily generate agreement/disagreement opinions from the participants. Table 3 presents the Likert statements. These were arranged randomly. The last three variables of the survey (PEU, PU, and IU) were deduced from the *Perceptions* and *Intentions* dimensions of the Method Evaluation model (MEM) for IS evaluation methods (Moody, 2003).

The last three items of the package were grouped into a spreadsheet to facilitate the data collection from the participants.

3.3 Tasks

During the experiment, the participants received a seminar about quality in modelling languages. Afterwards, a scenario was presented where multiple modelling languages are employed in an IS project. The participants evaluated the quality in this scenario using any other quality evaluation approach and the MMQEF method. The participants could choose between 6C Goals (Mohagheghi, 2009), PoN (Moody, 2009), SEQUAL (Krogstie, 2012a), and other criteria suggested by them (such as the result of combining principles of the identified frameworks).

Figure 1 presents the tasks performed in the experiments with their associated duration. Four sessions were required in order to work with all of the participants. Each session had different participants in accordance with their availability for the experiment and the dates of the Master'courses. The sessions were as follows:

S1: 24 participants, April-June 2016.
S2: 8 participants, November-December 2016.
S3: 11 participants, October 2017.
S4: 7 participants, June 2019.

Each session was performed in an academic location to facilitate the access of the participants to scientific databases and other resources required in the evaluation of quality for the modelling scenarios proposed in Section 3. This location also facilitated the face-to-face support between the researchers and the participants. The risks about situations that could affect the experiment were successfully addressed by the researchers. All of the participants performed the validation under the same conditions (a computing laboratory, internet access, modelling tools, and all of the experimental material described in Section 3.2).

Table 3 Likert sentences associated to MEM variables (the *Perceptions* and *Intentions* dimensions)

MEM variable	ID	Likert question
PEU	L1	MMQEF is easy to understand.
PEU	L2	It is easy to use MMQEF to detect quality issues in modelling languages and models.
PEU	L3	MMQEF is useful for detecting quality issues in modelling languages and models.
IU	L4	You would use MMQEF in later scenarios of quality assessment in models and languages.
PU	L5	The use of MMQEF allows problems in software engineering and information systems to be addressed using conceptual models.
PU	L6	MMQEF is aligned with the principles of the MDE paradigm (i.e. quality is evaluated from the MDE perspective).
IU	L7	From this experience, the evidence of quality at the level of modelling languages and models will be important in your further software engineering and/or information systems projects.
PU	L8	MMQEF allows relevant considerations for addressing a project under the model-driven paradigm to be identified.
PEU	L9	MMQEF provides a practical method to identify quality issues in projects developed under the model-driven paradigm.
PU	L10	The taxonomy that is used in MMQEF supports the construction of an information system using conceptual models, and it also considers the conceptual levels where models can be placed (e.g. from the *organization* level to the *implementation* and *deployment* levels).
PU	L11	The classification of modelling languages and modelling elements is useful in finding quality issues in model-driven projects.
PU	L12	The inferences proposed by MMQEF contribute to identifying quality issues in MDE projects.

3.4 Hypotheses, parameters, and variables

The following hypotheses were defined for the experiment:

H0: Compared to alternative methods for evaluating quality in MDE (e.g. SEQUAL, PoN, and 6C Goals), participants do not perceive the applicability of the MMQEF method for finding quality issues in MDE projects.
Ha: The applicability of MMQEF is perceived by the users of the method.

The independent variable that was defined for the experiment was the *method* (its application) to evaluate quality in MDE contexts. Table 4 presents the variable with its possible associated values. The other approach could be one of the following options: the 6C Goals, the Physics of Notations (PoN), the SEQUAL framework, or any personal criteria applied by the subject to perform an evaluation procedure. The first three frameworks were taught as part of the seminar that we provided.

Some dependent variables were identified. Table 5 describes the variables, where the first six variables were obtained from the application of each method according to

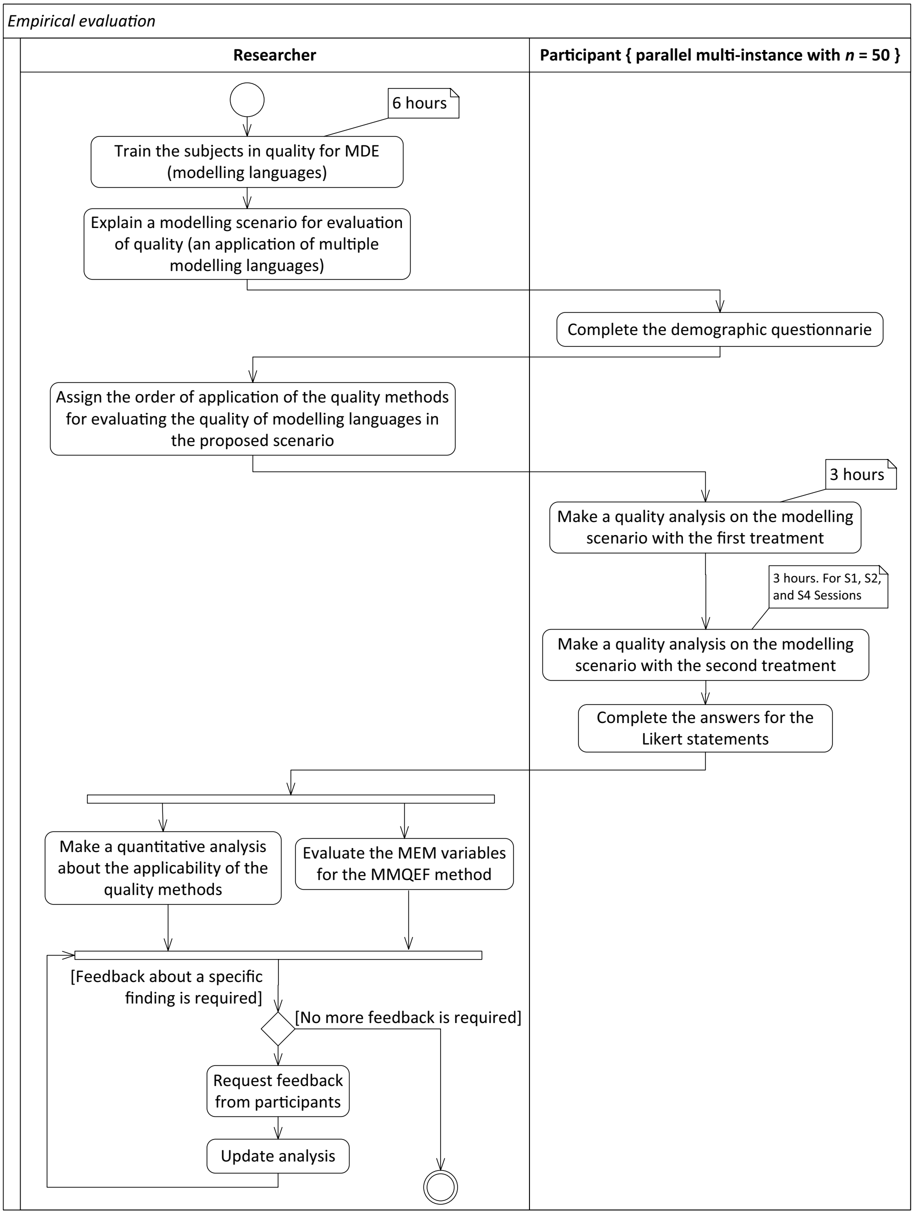
Empirical evaluation
Researcher
Participant { parallel multi-instance with n = 50 }
6 hours
Train the subjects in quality for MDE (modelling languages)
Explain a modelling scenario for evaluation of quality (an application of multiple modelling languages)
Complete the demographic questionnarie
Assign the order of application of the quality methods for evaluating the quality of modelling languages in the proposed scenario
3 hours
Make a quality analysis on the modelling scenario with the first treatment
3 hours. For S1, S2, and S4 Sessions
Make a quality analysis on the modelling scenario with the second treatment
Complete the answers for the Likert statements
Make a quantitative analysis about the applicability of the quality methods
Evaluate the MEM variables for the MMQEF method
[Feedback about a specific finding is required]
[No more feedback is required]
Request feedback from participants
Update analysis

Fig. 1 Summary of the tasks that were involved in the experimentation

the treatments defined in Table 2. The fourth and fifth variables refer to *inferences*, i.e. conclusion(s) that participants eventually might deduce when they applied a quality method (e.g. modelling language A is better than modelling language B for addressing a system concern). The last three variables are from the MEM model.

Table 4 Expected treatments for the independent variable

Factor	Treatments
Method	Application of any other approach Application of MMQEF

3.5 Analysis procedure

The *Sign test* approach was used to analyse H_0 and H_a, (see Section 4.2). To analyse the hypotheses, we assume that the behaviour of the independent variable (*application of the method*) has a *binomial distribution* because the following four conditions were present:

- The number of observations was fixed.
- Each observation was independent.
- Each observation represented one of two possible outcomes (*success* or *failure*).
- The probability of *success* (p) was the same for each outcome.

The distribution is described as $X \sim B(n, P)$ where $n =$ number of analysed observations and $X =$ number of successful (or positive) answers. For this analysis $P = 0.5$ and $p - value = 0.05$. The null hypothesis is accepted when ($P \leq 0.5$); otherwise ($P > 0.5$), it is rejected. Assuming that quality methods can be comparable through the T, DoU, QIF, NQI, QID, and NI dependent variables, we tested H_0 and H_a with $X \sim B(n, P)$ and $P(X \geq x)$.

Each distribution has its associated dependent variable and the respective values for computing it, including the obtained x value (positive answers). Successful or positive answers (x value) were detected for the cases when the subject assigned a positive response to the MMQEF method in each one of the six related variables:

- $T_{MMQEF} < T_{AnotherMethod}$
- $DoU_{MMQEF} > DoU_{AnotherMethod}$
- $QIF_{MMQEF} = YES \wedge QIF_{AnotherMethod} = NO$
- $NQI_{MMQEF} > NQI_{AnotherMethod}$

Table 5 Dependent variables identified for the validation

Description	Acronym	Values
Time of application of the method	T	[0 ...)
Degree of understanding of the method	DoU	[0 ...100]
Were quality issues found?	QIF	YES / NO
Number of quality issues derived by each subject from the proposed scenario	NQI	[0 ...)
Were quality inferences deduced?	QID	YES / NO
Number of *inferences* deduced from the application of the method	NI	[0 ...)
MEM Perceived Ease Of Use for MMQEF (PEU)		[1 ...5]
MEM Perceived Usefulness for MMQEF (PU)		[1 ...5]
MEM Intention To Use for MMQEF (IU)		[1 ...5]

- $QID_{MMQEF} = YES \wedge QID_{Another} = NO$
- $NI_{MMQEF} > NI_{AnotherMethod}$

Negative answers were those in which MMQEF received a low or equal score (tie). In addition, a direct analysis with the Likert values obtained was used to analyse the MEM variables (see Section 4.2.1)

3.6 Execution - deviations

For sessions *S3* and *S4* (Section 3.3), we introduced an additional question for the participants in order to determine the reason(s) why they selected the quality method that was employed in the experiment as an alternative to MMQEF. We provided some non-exclusive reasons for this question:

- The *ease of use* of the approach.
- The examples found that are formulated for the approach.
- The *suitability* of the approach regarding the application domain.
- The documentation of the approach.
- *I think it is the best option.*
- *The approach can be performed in the allotted time (3 hours).*
- *No particular reason. The choice was random.*
- Another unspecified reason.

The participants that were previously in sessions *S1* and *S2* were also contacted by email in order to ask them the same question.

In addition, for the *S3* session, we assign only one quality evaluation method to each participant in a *paired-comparison* approach, in order to detect possible advantages that quality methods could individually provide.

4 Analysis

4.1 Descriptive analysis

Before the interaction with MMQEF and the other quality evaluation frameworks, the participants were asked whether they knew some key terms that are employed in the MMQEF method. If an affirmative answer about knowledge of these terms was given, the participants were also asked about the application of these concepts in their immediate contexts.

Table 6 summarizes the percentages for the knowledge and application of the terms involved. These percentages indicate a positive trend, which could infer an appropriate use of MMQEF due to the importance of those terms in the methodo-logical specification of the MMQEF method. In addition, for the same questions (knowledge and application), if the participants gave an affirmative answer, they were asked to indicate the associated level for both of them using a Likert scale from 1 to 5, with 1 indicating the lowest level of knowledge/application and 5 indicating the highest level. Table 7 presents the levels of knowledge/application obtained for the participants.

For the participants, there was a dramatic change in their initial perception of familiarity with the MMQEF key terms. While the percentages in Table 6 show a positive

Table 6 Percentages of knowledge and application of MMQEF key terms

MMQEF key terms	Participants (n=50)	
	Do you know it?	If you do, do you apply it in your professional tasks?
MDD	86.00%	48.00%
MDA	74.00%	37.84%
MDE	76.00%	52.63%
Conceptual modelling	84.00%	64.05%
Metamodelling	82.00%	43.90%
Model transformation	74.00%	43.24%
Transformation languages	72.00%	33.33%
Traceability	70.00%	60.00%
Abstraction level	80.00%	67.50%
Viewpoint	48.00%	58.33%
View	58.00%	58.62%
Model-driven technical environments	78.00%	51.28%
Model-driven tools	82.00%	60.98%

trend about knowledge and application, the associated levels in Table 7 present moderate (and relatively low) behaviour. This indicates a limited comprehension of the terms, which eventually affected the full understanding and use of the MMQEF method (see Section 4.2). The concept of *conceptual modelling* can be highlighted for its association with the data representation for relational databases.

After the characterization of the MMQEF terms, the participants were asked about their knowledge and use of modelling languages and domain-specific languages (DSLs).

Table 7 Resulting levels of knowledge and application of MMQEF terms for the participant

MMQEF key terms	Participants (n=50)					
	Knowledge level			Application level		
	Mean	Median	Mode	Mean	Median	Mode
MDD	2.62	3.00	3.00	2.73	2.00	2.00
MDA	2.42	2.00	2.00	2.44	3.00	3.00
MDE	2.95	3.00	4.00	2.92	3.00	4.00
Conceptual modelling	3.08	3.00	3.00	3.50	3.00	3.00
Metamodelling	2.82	3.00	3.00	3.33	3.00	3.00
Model transformation	2.50	2.50	1.00	2.58	2.50	2.00
Transformation languages	2.39	2.00	1.00	3.00	4.00	4.00
Traceability	3.20	3.00	4.00	2.94	3.00	3.00
Abstraction level	2.80	3.00	2.00	2.81	3.00	3.00
Viewpoint	2.73	3.00	4.00	3.27	4.00	4.00
View	2.94	3.00	3.00	3.08	3.00	3.00

They were also asked about their intention to use these two approaches in their immediate contexts. Table 8 presents the results.

The population reported knowledge of modelling languages (94%). UML was the most popular modelling language (59.57%). BPMN was the second modelling alternative (44.68%), and other specific alternatives appeared, such as ER (6.38%), i* (4.26%), SySML, Flowchart and AADL (each with a percentage of 2.13%).

SQL was the DSL that was most known by the population (36.84%). XML was the second technical DSL (7.89%). Individually, some participants reported their knowledge of DSLs, such as HTML, R, MPI, and VHDL.

The above percentages are a consequence of the *convenience sampling* approach that was applied to select the participants. Despite their lack of background in the key terms, the familiarity of the participants with modelling languages made them appropriate participants for discussing quality issues in modelling languages and the posterior application of quality evaluation frameworks.

4.2 Testing of hypotheses

First, we detected an important tendency of the participants to choose a quality evaluation method with more prescriptive information and orientation about tasks, steps, procedures,

Table 8 Information about knowledge and use of modelling languages and DSLs from the participant

Element	Questions	Participants with YES answer	%	
Modelling languages	Do you know modelling languages?	47	94.00%	
	If you do, what do you use them for?	42	89.36%	
	If you do, what do you use them for	Options	Total	%
		For documentation	35	83.33%
		To generate code	17	40.48%
		To generate models	9	21.43%
		To communicate (e.g. to share ideas, to explain some concept)	27	64.29%
		For other purposes	6	14.29%
DSL	Do you know DSLs?	38	76.00%	
	If you do, do you use them?	35	92.11%	
	If you do, what do you use them for?	Options	Total	%
		For documentation	11	26.19%
		To generate code	24	57.14%
		To generate models	14	33.33%
		To communicate (e.g. to share ideas, to explain some concept)	12	28.57%
		For other purposes	9	21.43%

etc., to perform the quality evaluation. A total of 46.15% of the participants chose the PoN, and 38.46% chose the 6C Goals. Only one professional reported the use of the SEQUAL framework. However, in his reported data, there was clearly a conceptual confusion when he applied it.

In accordance with the analysis that was described in Section 3.5, we compared the application of the quality methods for each participant, identifying the cases in which MMQEF demonstrated an advantage over the other selected quality method (i.e. the x parameter of the Binomial Distribution).

When we processed the raw data, we detected *No Response (NR)* cases (i.e. empty responses) and *Tied Score* cases (i.e. situations where MMQEF and the other selected quality method have equal values for the same variable). Therefore, in order to determine the values to be compared with the Binomial Distribution, the value of n corresponds to the number of participants (with a sample of 39 participants which corresponds to sessions S1, S2 and S4) subtracting *No Response* and *Tied Score* situations. Due to this variation, the n values for T-, DoU-, QIF-, NQI-, QID-, and NI-dependent variables changes.

Table 9 presents the obtained results. For each dependent variable, the amount of successful application of MMQEF is indicated by the x parameter. *NR* and *Tied Score* cases are also reported. Regarding the analysis that was initially proposed in Section 3.5, we report the explicit *tied* cases in order to show the high number of those obtained cases (especially for QIF and QID variables) as a consequence of the first interaction of the participants with the quality evaluation methods that were proposed by the researchers. This finding does not affect the $P(X \geq x)$ analysis in Table 9.

The probability distributions presented in Table 9 oblige us to accept H_0. From the obtained results, we could make a *Type-I-error* (Wohlin et al., 2012a), attempting to justify the applicability of the MMQEF method without a pattern of positive distribution over the data.

The above resulting scenarios were an expected output; this is a consequence of verifying H_0 and H_a with six questions/metrics. In addition, the resolution of the metrics was based on the subjective criteria of each participant and their first use of the quality evaluation frameworks, including MMQEF. Thus, the overall results require a review of the complementary answers provided by the participants in each experiment in order to find evidence that adequately justifies the responses provided.

Initially, the *time used to apply the quality evaluation method* (T) was considered to be a metric to show the practicality of MMQEF when compared to existing methods. However, in the results obtained for this metric, there is no significant difference between the time of application for the MMQEF method and the time of application for the other methods used by the participants (including the *personal criteria* method). For the *significant difference*, we expected a difference of (at least) thirty minutes between the applications of the methods. However, 68.75% of the professionals reported times without any significant difference (i.e. equal times or times whose difference was less than thirty minutes).

Table 9 Binomial distribution results

Variable	MMQEF (x)	OTHER (X)	Tied score	NR	n	$P(X \geq x)$
T	10	15	13	1	25	0.88524
DoU	15	15	6	3	30	0.57223
QIF	5	5	28	1	10	0.62305
NQI	12	16	10	1	28	0.82754
QID	3	6	29	1	9	0.91016
NI	11	13	14	1	24	0.72937

Table 10 Comparison of dependent variables associated to the hypotheses

Dependent variables	MMQEF	Other Method
T average (minutes)	66.1	57.1
DoU average (%)	49.9	56.5
Number of participants who found quality issues (QIF)	30.0	31.0
NQI Average	1.6	2.1
Number of participants who reported inferences (QID)	28.0	31.0
NI Average	1.3	1.4

Table 10 presents the average value of the dependent variables involved for testing the two hypotheses. It shows that the values for QIF, QID, and NI variables are close. The reported average of quality issues that were found and the number of inferences that were reported are relatively low in accordance with the time that was given to the participants for working with each quality method, the supplementary support that was given to the participants, and the additional support that the participants found by themselves. Taking into account these averages, the efficiency of the quality methods (MMQEF and the others selected) is inevitably questioned.

Table 11 shows the comparison of dependent variables to the S3 session (see Section 3.6). Values for the dependent variables are close, which demonstrates no advantage of the quality evaluation methods when they were individually applied.

These findings may be a consequence of the lack of previous knowledge about the model-driven paradigm (as reported in Table 7) and the first interaction of the participants with methods to evaluate quality in MDE. Reported methods for evaluating quality in MDE require great cognitive effort for their understanding and applicability. The previous conception of *quality* of each participant also influenced the performance of the participants during the experimentation. In addition, quality methods are not directly comparable with each other by using their purposes, scopes, procedures, and conception of quality.

Due to the low values of the NQI averages that are presented in Table 10, we took advantage of Table 1 (which presents the expected quality issues for the participants that we previously considered in Section 3) in order to determine the applicability of quality methods through a review of the responses from the participants indicating whether or not they found similar issues to those of Table 1. Table 12 presents each projected issue with the number of participants that reported similar issues to the ones projected.

Table 12 also presents other quality issues reported by the participants, which can be classified according to identified categories for issues. Other quality issues reported by the

Table 11 Comparison of dependent variables associated to the S3 Section

Dependent variables	MMQEF	Other Method
DoU average (%)	36	44.2
Number of participants who found quality issues (QIF)	3	6
NQI Average	4.3	4.3
Number of participants who reported inferences (QID)	3	5
NI Average	1.7	1.2

Table 12 Number of participants who reported issues similar to those that are projected in Table 1

ID	Quality Issue	Number of related reported issues	Source
QIPR01	There was no traceability from models until code.	4	MMQEF
QIPR02	There was no automatic code generation.	1	MMQEF
QIPR03	Decoupling between organizational modelling and system modelling.	1	MMQEF
QIPR04	Misalignment between the modelling languages used and the purposes of modelling.	4	MMQEF
QIPR05	Excessive stereotyping of the UML modelling language.	7	MMQEF, PoN, personal criteria
QIPR06	Excessive adaptation of languages to model business concerns.	4	MMQEF
QIPR07	Lack of suitability analysis of modelling languages.	9	MMQEF, 6C, personal criteria
QIPR08	Lack of coverage of modelling languages. Some IS issues were not covered by modelling languages (e.g. data, interaction, architectural decisions).	12	MMQEF, 6C
QIPR09	There was no distinction of the purpose of the resulting models (i.e. there were models to communicate ideas, to automate processes, to make systems), but these purposes were not explicitly addressed.	2	MMQEF, SEQUAL
QIPR10	There was no integration between modelling languages.	4	MMQEF, 6C
QIPR11	Poor support for the deontic level (for organizational and system purposes).	1	Personal criteria
QIPR12	Lack of adequate tool support.	0	
QIPR13	Lack of expressiveness of the modelling languages.	6	MMQEF, personal criteria, PoN
QIPR14	Lack of communication abilities for the performed modelling.	4	6C, PoN
Other quality issues (from professionals)	There are redundant elements in the modelling languages under evaluation.	1	MMQEF
	Quality issues related to symbols.	1	PoN
	Changeability issues.	2	6C
	Specific quality issues.	5	SEQUAL, 6C, personal criteria, PoN

participants in Table 12 are specific quality issues without a common category for grouping them (i.e. a category that differs from the quality methods) due to their specificity (5 issues reported in other methods by participants).

Because of the low differences obtained in the value of the variable, and the number of reported issues that are reported in Table 12, the potential applicability of MMQEF can be inferred considering its first usage in conjunction with other quality evaluation methods for MDE.

An analysis was also performed to determine the influence of the quality frameworks on each other as a consequence of the *Paired-comparison design* (defined in Section 3.1). To do this, we compared the identified favorable cases for MMQEF in each of the values obtained by the T-, DoU-, QIF-, NQI-, QID-, and NI-dependent variable, regarding the application of methods defined in Table 2.

Table 13 summarizes the identified favorable cases for MMQEF regarding the treatments of Table 2. There is similar behaviour in the value of favorable cases regarding the distribution design. For the T and DoU variables, more cases were favorable for MMQEF when the participants started with other methods; however, for the QIF, NQI, and NI variables, a greater number of cases were reported when the participants started with MMQEF. For the QID variable, there is not a significant difference in the obtained values of favorable cases. Because of the divergence in the behaviour of the *paired-comparison* design, there is not a pattern of influence between methods regarding the treatments of Table 2. Therefore, there is no evidence of the influence of the quality frameworks on each other (i.e. MMQEF on the others and vice versa).

4.2.1 Analysis of the MEM variables for MMQEF

One of the main risks of this empirical evaluation is the use of new methods that are absolutely unknown to the participants. In (Wohlin et al., 2012a), Wohlin et al. discuss the risks involved when new methods are tested; they specifically consider issues related to the consistent application of previous methods and their influence on existing methods

Table 13 Comparison of favorable cases for MMQEF regarding the paired-comparison design

Expected success answer for MMQEF	Distribution (starting with)	Number of favorable cases for MMQEF variables
$T_{MMQEF} < T_{AnotherMethod}$	MMQEF	3
	Other method	7
$DoU_{MMQEF} > DoU_{AnotherMethod}$	MMQEF	5
	Other method	9
$QIF_{MMQEF} = YES \wedge QIF_{AnotherMethod} = NO$	MMQEF	5
	Other method	0
$NQI_{MMQEF} > NQI_{AnotherMethod}$	MMQEF	10
	Other method	2
$QID_{MMQEF} = YES \wedge QID_{Another} = NO$	MMQEF	3
	Other method	0
$NI_{MMQEF} > NI_{AnotherMethod}$	MMQEF	8
	Other method	3

when new ones are learned. A clear example of such a validation can be found in (Panach et al., 2015), where an experiment was performed to compare a traditional software process development with a model-driven development process.

However, in the design of this validation, the main challenge is the lack of previous interaction with the quality frameworks for MDE. This was the first time that the participants confronted quality issues in MDE, and, therefore, their first time recognizing and applying the frameworks involved (including MMQEF).

Although, each participant in the S1, S2, and S4 sessions applied two quality evaluation methods (freely selecting one of them), the entire population (including S3 session) had not considered the presence of quality issues in modelling languages. Therefore, they did not know frameworks or methods for addressing quality evaluation procedures at the model-driven level. This was evident despite the percentage of the participants who reported previous knowledge and skills with model-driven technical environments, use of modelling languages, use of domain-specific languages, and metamodelling (see Section 4.1).

For this reason, a Likert survey approach was applied to validate the specific MEM dimensions of the MMQEF (i.e. the *Perceived Ease Of Use*, the *Perceived Usefulness*, and the *Intention To Use*) that are described in Table 5 of Section 3.2. The survey was applied mainly to determine if MMQEF could have been influenced (and affected) by the application of other quality frameworks selected by the participants, complementing the finding of Table 13 by the use of the last three dependent variables that are defined in Table 5. Table 14 summarizes the results obtained in the Likert sentences for the participants.

Section 7 has the URL for accessing the data and the comparative figures associated to each Likert sentence.

Table 14 also presents the Cronbach's α that were obtained for the Likert survey of the professionals ($\alpha = 0.87897523$). Since the obtained values are greater than the expected value for this test (0.7), the reliability of the Likert survey and their internal consistency is confirmed.

Since we considered *positive responses* to be 4 (Agree) and 5 (Strongly agree) on the Likert scale, there is enough evidence to recognize the positive responses to the MEM dimensions for MMQEF. Of all of the Likert sentences, the professionals gave the greatest

Table 14 Summary of the Likert responses of the participant for MMQEF

Likert	Likert scale (Cronbach's $\alpha = 0.87897523$)							
Question	1	2	3	4	5	Mean	Median	Mode
L1 - Ease of understanding MMQEF	4	0	18	18	4	3.41	3.5	3
L2 - Ease of using MMQEF	4	0	15	19	6	3.52	4	4
L3 - Usefulness of MMQEF	1	1	14	14	13	3.77	4	3
L4 - MMQEF further intention of use	2	0	7	16	17	3.91	4	5
L5 - Usefulness of MMQEF in SE and IS problems	1	1	6	20	15	4.00	4	4
L6 - Alignment of MMQEF with MDE principles	1	0	14	12	16	3.89	4	5
L7 - Further importance of quality at the MDE level	1	1	8	15	18	4.02	4	5
L8 - Considerations of MDE projects addressed with MMQEF	1	0	4	22	16	4.11	4	4
L9 - Practicality of MMQEF	2	3	10	22	7	3.66	4	4
L10 - Usefulness of the taxonomy	1	1	6	15	21	4.23	4	5
L11 - Usefulness of classification of modelling languages	0	0	5	20	18	4.20	4	4
L12 - Usefulness of the MMQEF inferences	0	3	11	18	12	3.89	4	4

number of *3* values (neither agree nor disagree) to the Likert 1 statement (*ease of understanding MMQEF*). This is a consequence of the first interaction of the professionals with the taxonomy proposed in the Zachman framework, which requires an initial cognitive effort for the use of its bi-dimensional structure and rules.

The obtained data demonstrate a trend towards the selection of the 4 and 5 Likert levels in the participants. To confirm this, we apply a *quartile analysis* for the MEM variables. For each participant, the sum of the values of the Likert sentences associated to each MEM variable was calculated (i.e, the Likert sentences 1, 2, 3, and 9 for the *Perceived Ease Of Use* variable, the Likert sentences 4 and 7 for the *Intention To Use* variable, and the Likert sentences 5, 6, 8, 10, 11, and 12 for the *Perceived Usefulness* variable).

Figure 2 presents the resulting quartile analysis. This figure shows the trend of the distribution of the samples to higher values that were expected for each MEM dimension, i.e, *20* for PEU (four Likert sentences), *10* for IU (two Likert sentences), and *30* for PU (six Likert sentences). However, the performance of PEU demonstrates that the *Ease of Use* feature of MMQEF is directly influenced by the comprehensibility of the rules of the method and its associated evaluation tasks.

4.3 Summary

In summary, the findings of this section can be described as follows:

- Key terms for the MMQEF method (and the model-driven paradigm itself) are not properly appropriated by the participants. This influences the application of MMQEF for evaluating quality in model-driven scenarios.

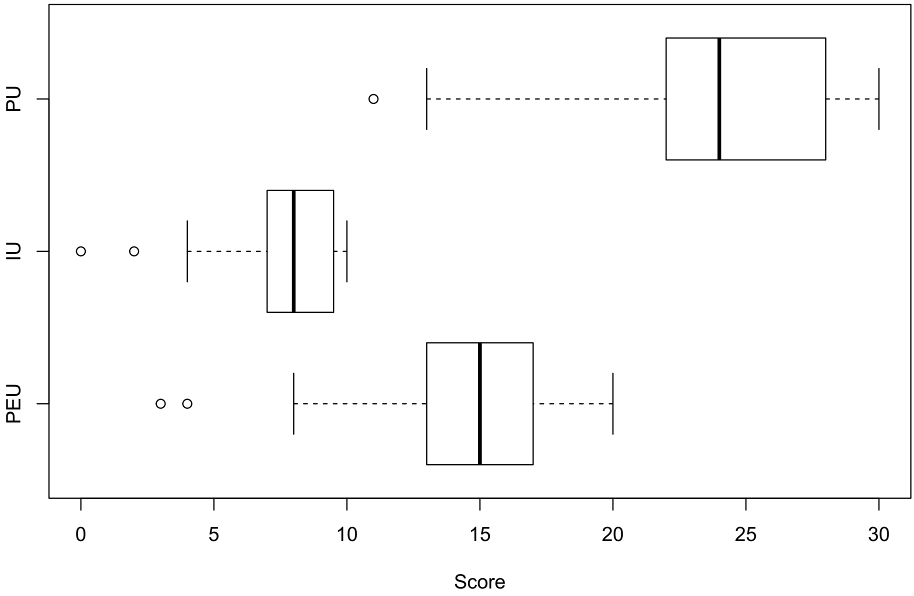

Fig. 2 Quartile analysis for the Likert categories

- The quantitative analysis of the application of the quality methods does not indicate an important difference between MMQEF and the other methods. Instead, the obtained results question the efficiency of the performance of the quality methods.
- Quantitative analysis approaches reveal a potential applicability of MMQEF regarding the other quality methods under similar conditions of practice. This was deduced from the values of dependent variables that were freely reported by the participants.
- The results from the Likert sentences demonstrate that the participants agree and recognize the dimensions of the MEM model (i.e. the *Perceived ease of use*, *Perceived usefulness*, and *Intention to use*) for the MMQEF method.

5 Discussion

It is widely recognized that quality is an intrinsic property of artifacts in engineering itself. However, there is no common consensus regarding quality beyond its explicit manifestation and management. Instead, multiple definitions and proposals about quality have been identified, each of which is valid based on the specification of quality that is addressed.

The validation processes in frameworks for MDE are highly specific. Concrete scenarios are required to be able to integrally validate quality frameworks in MDE and to demonstrate their feasibility and efficiency. Quality frameworks for MDE could be mutually exclusive depending on the concept of quality being addressed. Due to these conceptual divergences, the methods may not be mutually comparable with each other.

Each framework for quality evaluation in MDE provides specific purposes and advantages based on its specific concept of quality. As a consequence of the broad definition of quality for MDE, several types of quality can be proposed and their purposes can be justified. The challenge here is the systematic integration of these frameworks to promote the adoption of the MDE paradigm by focusing on quality and its associated implications.

5.1 Threats to Validity

For this experiment, the validity of the results is critical due to the need to review the obtained data and extract evidence about the positive applicability of MMQEF. The participants voluntarily enrolled in the experiment, and they were free to leave it at any time. The participants also signed a consent form. The data were integrally managed as it was reported by the participants.

The method evaluation model (MEM) for the evaluation methods of information systems (Moody, 2003) was also intentionally applied to validate the potential applicability of MMQEF through an approach that is proposed for evaluating information system methods. Associated items for MEM perception variables (i.e. the *perceived ease of use*, the *perceived usefulness*, and the *intention to use*) were checked by using approaches to determine the reliability and internal consistency of the obtained responses (Section 4.2.1).

Taking advantage of the checklist of validity threats that was defined in (Wohlin et al., 2012b), Tables 15 and 16 present a detailed analysis of the threats that were detected for this empirical validation and the corresponding strategy for addressing them. Overall, we acknowledge having suffered several threats that were difficult to eliminate completely; however, we applied mitigation strategies to minimize the impact of these threats.

Table 15 Analysis of *validity threats* for the experiment (Part I)

Validity category	Threat	State	How we addressed it in the empirical validation.
Conclusion	Low statistical power	Addressed	The statistical evidence was respected by following strict protocols for applying the statistical analysis. True patterns from the data are evident. The statistical patterns found obligate us to apply complementary tasks from qualitative research to support the applicability of MMQEF.
	Violated assumptions of statistical tests	Addressed	Statistical assumptions were preserved in the test of hypotheses (binomial distribution) and the analysis of Likert sentences (through a quartile analysis and a Kruskall-Wallis test). For those analyses, we use parameters as they are commonly reported in the literature. In addition, for the Likert analysis, we apply two recognized configurations for discussing a found pattern about the similarity of distributions for a MEM variable.
	Fishing	Addressed	Quantitative data were enough for analysing the evidence of the applicability of MMQEF. No another data (e.g. qualitative data) were required for the analysis. The report of *tie scores* was proposed due to the number of detected cases in the delivered responses.
	Error rate	Addressed	Values of the significance level variables that were employed during the analysis (i.e. values for α and $p - value$) were determined from a literature review about the settings for experiments of this kind that are commonly suggested, and also from expert judgment from statistical advisors. No multiple analyses were conducted in the experiment.
	Reliability of measures	Addressed	The instruments of the validation employed were double checked. In addition, the validation was applied in four sessions in accordance with the availability of the participants. For all the sessions, the obtained outcome was similar. No differences in the behaviour of the results obtained were found.
	Reliability of treatment implementation	Addressed	The treatments in Table 4 were carefully applied in S1, S2, and S4 sessions of the validation.
	Random irrelevancies in experimental setting	Addressed	No elements outside the experimental setting were presented in each session of the experiment.
	Random heterogeneity of subjects	Addressed	To guarantee the heterogeneity of the participants in the validation, 50 participants with verified expertise in real software development projects were invited. The participants were demographically analysed (Section 4.1), no evidence of features in the sample that represent threats for the result was detected.
Internal (single group threat)	History	Addressed	Despite the four sessions that were required in the validation, no effects by the execution of the validation at different times were reported. The participants worked once on the experiment. No repeated test was required in the design of the experiment.
	Testing		

Table 15 (continued)

Validity category	Threat	State	How we addressed it in the empirical validation.
	Maturation	Partially addressed	The researchers addressed it by the treatments defined in Table 2. However, there was an initial cognitive load for interacting the first time with quality methods for MDE. Eventually, this load affected the performance of the participants. Taking into account the number of participants, an alternative for managing the maturation threat was to assign the application of only one method to each participant. However, this decision could have affected the statistical power due to the low results obtained. For experiments of this kind, it is complex to involve subjects without affecting the quality of the training phase.
	Instrumentation	Addressed	The form that was designed to collect data from the participants has a simple and practical design to facilitate the interaction of each participant with the instrument. No reports about the cognitive complexity of the form were presented.
	Statistical regression	Addressed	The participants were chosen by a convenience sampling. No additional classification tasks were applied to the participants.
	Selection	Addressed	Due to the previous knowledge about modelling, quality, and MDE topics that were required of the participants, a random selection was not viable.
	Mortality	Not applicable	The participants were free to leave the experiment at any time. However, none of the participants dropped out of the validation.
	Ambiguity about direction of causal influence.	Not applicable	During the evaluation of quality in a modelling project using some proposed methods, the participants must apply subjective criteria to determine the cause/effect relation of the quality issues found. The available information of the methods that was provided by the researchers supports the individual analysis. The ambiguity is implicitly addressed by the application of the quality evaluation methods.
	Diffusion or imitation of treatments	Partially addressed	One case of an imitation in the treatment was detected in two professional participants. However, this did not influence the overall performance of this population. Instead, this finding was carefully managed to analyse the application of the methods that was reported by the participants involved. The imitation of the treatment did not impact the application of the methods of each participant.

5.2 Inferences

The following items generalize the findings of the experiment:

5.2.1 The selection of practical methods.

Clearly, as we reported at the beginning of Section 4.2, the participants searched for quality methods that helped them to make the quality evaluation in the easiest and most practical way. The available supporting material for the quality frameworks influenced the selection of the participants. This was a disadvantage for MMQEF because it is a work-in-progress

Table 16 Analysis of *validity threats* for the experiment (Part II)

Validity category	Threat	State	How we addressed it in the empirical validation.
Construct	Inadequate preoperational explanation of constructs	Addressed	To avoid a lack of clarity in the participants with new theories about quality evaluation methods for MDE, the experimental material (Section 3.2) was taught in accordance with didactical strategies that facilitate and promote the interaction of the participants with these new (and unknown) theories.
	Mono-operation bias	Partially addressed	For the experiment, it was not possible to implement multiple versions of quality modelling scenarios for applying quality methods for MDE due to the complexity of each scenario and the several quality issues that could be reported from the subjective analysis of each participant. Multiple modelling scenarios impact the results about the applicability of methods. Specific values of the independent variable were employed for specific scenarios of applicability. These values and scenarios could limit the demonstration of the quality evaluation for MDE. Therefore, to address this in further experiments, we propose validating specific features of the quality methods for MDE with multiple experiences (Section 5.3).
	Mono-method bias	Addressed	Multiple (and complementary) measures were used in the experiment to evaluate the applicability of the methods and their potential use. Multiple resulting observations from participants are from the subjective interpretation and application of quality methods. Measures have behaved in accordance with the theoretical expectation of the researchers.
	Confounding constructs and levels of constructs	Partially addressed	The relationship between the previous knowledge of the participants and the obtained performance of the methods was explicitly reported (Section 4.2). In addition, the training about the quality methods placing emphasis on specific features (including MMQEF) could affect the results of selection and applicability of methods in the participants.
	Hypothesis guessing	Addressed	There was a consensus about the purpose of the experiment and the applied treatments. No guesses about the purpose were detected. Prior to the experiment, the researchers had identified and removed any possible guessing source for the experiment, e.g. the relationship between the performance and any qualification in the case of the Master's students.
	Restricted generalizability across constructs	Addressed	The results obtained can be considered in similar scenarios of experimentation about quality methods for MDE.
	Experimenter expectations	Partially addressed	We avoid any influence of the researchers favorable to MMQEF. This was done by contrasting its applicability with regard to the quality methods that were previously proposed in the MDE literature. Although the researchers had expectations about the potential applicability and potential use of the MMQEF method, during the sessions of the experiment, any attempt to influence opinions of participants was avoided. The support for the participants was carefully provided to prevent any opinion (and induction) favorable to MMQEF. A neutral role was assumed by the researchers even though they had positive expectations for MMQEF.

Table 16 (continued)

Validity category	Threat	State	How we addressed it in the empirical validation.
External	Interaction of selection and treatment	Addressed	A suitable population for the experimentation was convened by applying a convenience by sampling approach (Section 3.1). However, the characteristics of the population do not show any signs of a possible influence of their background on the results of any specific treatment (see the demographic analysis in Section 4.1).
	Interaction of setting and treatment	Addressed	The researchers were especially careful to provide representative material for the participants, including modelling scenarios for applying quality methods, in accordance with the previous knowledge of the participants (i.,e, the UML-BPMN-Flowchart-SPEM scenario for professionals (Section 3)).
	Interaction of history and treatment	Not applicable	No effects for the days and times of application of the experiment were reported.

and it does not have a lot of related publications. The selection of other methods based on their associated supporting material was evident despite the supporting material for MMQEF that we provided. The participants searched for specific examples of applications of the quality methods by reviewing their derived publications.

The selection of more practical frameworks is a consequence of the formulation of the desirable properties that represent quality in modelling languages, despite the lack of the proper description of procedures about how to determine the fulfillment of these properties. For example, with the PoN framework, all of the participants were warned about the lack of a systematic application for this method. The participants were also warned that it was not until 2016 that authors other than the original authors had proposed guidelines to address this application (da Silva Teixeira et al., 2016).

5.2.2 Detected reasons for choosing quality methods

Table 17 summarizes the participants' reasons for choosing the alternative quality methods to MMQEF (Section 3.6). The table shows that 62.5% of the professionals stated that the *easiness* of the approach was the main reason for choosing it.

The participants of the *S1* and *S2* sessions were contacted by email, in which we asked them about the reasons for choosing the alternative quality method that was used in the experiment. To date, seventeen professionals have answered the email. Twelve of these professionals also stated the *easiness* of the approach as the main reason for choosing it. Some of these professionals indicated that their selection was also based on a relationship between the easiness of the selected method with the time allotted for working with the alternative method (seven professionals), the examples of application found of the selected approach (six professionals), and the associated documentation of the methods (five professionals).

Table 17 Reported reasons for choosing the alternative method during the experiment

Proposed reasons	Participants
The easiness of the approach.	11
The examples found that the approach had.	4
The suitability of the approach.	3
The documentation of the approach.	4
I think it is the best option.	4
The time required.	3
No particular reason. The choice was random.	1
Another reason (unspecified).	0

5.2.3 *Representations* as an important source for quality evaluation procedures.

An important finding that has been derived from the analysis of the independent variable (the application of the quality method) was the identification of the sources that were used by the participants to perform the quality assessment of the given models and modelling languages. In the application of each method (MMQEF and the alternative), each subject was queried about which sources of information he/she used to find quality issues. Table 18 presents the responses obtained.

The participants reported the use of *representations* (i.e. instanced models that are expressed in diagrams and/or textual blocks) associated to the modelling languages as the main sources for applying the quality frameworks and making quality assessments. Representations are the result of a cognitive interpretation of the users of the languages about the possible use and application of the modelling languages. Thus, quality issues are the result of an interaction among the participants with the modelling languages under analysis. Quality issues were detected from the perspective of the participants as the final users of the modelling languages. There is no evidence of quality issues from a *modelling language analyst* perspective (i.e. the role that creates, designs, or proposes a language for modelling a specific concern).

Although the quality frameworks provide guidelines for the correct use of the modelling languages, the application of languages by their associated representations (i.e. the possible instanced models that could result for a modelling language from an final-user perspective) is an important source of problems perceived by the final users of the languages with respect to *quality in use*. There is a relation between diagrams (instanced models) and the selection of quality methods that work prescriptively with the information extracted from these diagrams.

Table 18 Sources of information for detecting quality issues reported by the participants

Sources of quality issues	MMQEF	Others
Metamodel (grammar)	14	14
Representation	21	31
Complementary info (e.g. the use of a modelling tool)	7	11
Other sources	4	1

5.2.4 The need for a modelling context

A key finding that was detected in the experiment is the need to consider an explicit modelling scenario upon which the evaluation of the modelling languages could be done. We presented an illustrative scenario to the participants (Section 3.2). It is clear that all of the participants required a specific context to identify and report quality issues. The context was used as a pivot to detect quality issues. The participants did not compare languages and did not apply a quality method without the modelling context. This acted as a conceptual framework that helped the participants to contrast the scope of the modelling languages under analysis.

5.2.5 Perceived independence of the quality proposals

A clear trend in the obtained results was the application of quality methods as isolated frameworks to perform quality analysis on modelling languages. For the independent variable (*application of the method*), the participants were induced to use MMQEF and any other quality method. This second method was freely selected by participants without any intervention by us.

None of the participants proposed an integration of two or more methods to make quality assessments. All of the participants chose and applied quality methods individually; they were not concerned about any possibility of integrating methods. This indicates that quality methods were used as inductive tools to understand quality issues at the modelling level and to find them based on the quality concept proposed by the selected framework.

5.3 Lessons learned

5.3.1 The improvement of the procedure for making inferences in MMQEF

MMQEF provides explicit activities to formulate inferences about the application of the modelling languages and their classification in the taxonomic structure of the method. The taxonomy considers modelling realities from business to technical levels. For this reason, we consider that inferences can be easily detected from the location of the modelling elements and the artifacts regarding the information that can be captured by the cells of the taxonomy.

However, an important result of the experiments is that there is a need to improve the MMQEF guidelines in order to make inferences from the application of the framework. The obtained evidence demonstrates that the inferences are personal conclusions of the participants about the method itself. Thus, more practical and methodological orientation is required so that MMQEF users can detect the consequences of the classification of modelling languages and artifacts in accordance with the perceived application and scope.

5.3.2 The improvement of the documentation for MMQEF

There was an evident disadvantage for the MMQEF method with regard to its associated documentation and supporting material, especially for examples of application. The findings described in Section 5.2.1 demonstrate the preference of the participants for methods that provide explicit examples and documentation about the application of quality methods in specific scenarios. Documentation with prescriptive steps and guidelines for performing evaluation procedures with MMQEF must be developed in order to improve the interaction of the method with its potential target public (i.e. the modelling language analyst and the designer as well as the final user of the modelling languages).

5.3.3 Improving the process of selection and characterization of participants

Previous intermediate/advanced knowledge of software engineering concepts (especially technical knowledge) does not guarantee the suitability of the participants, as described in Section 4.1. The application of quality methods could be affected by previous conceptions from technical levels of software development projects, in which quality is based on the source code of programming languages and the progress of development teams. Further experiments must consider the *degree of appropriation* of MDE concepts and supporting technologies by the participants. Several configurations or scenarios for experimentation can be obtained from the identified MDE appropriation of the invited participants.

5.3.4 Validating specific features of quality methods for MDE individually instead of all together

The evaluation of the performance and applicability of quality methods for MDE can be affected by the complexity of their underlying theory. Evaluating all of the features of quality methods in a single experiment requires dense material for the experimentation with participants, and, therefore, complex procedures in the experimentation, especially for experiments of one session. Experiments about specific features of quality methods could be a more practical strategy for identifying and characterizing the effectiveness of these features. In addition, the focus on specific features of quality methods facilitates the eventual formulation of experiments for comparing features from different quality methods with similar principles and intentions.

6 Conclusions

6.1 Summary

The validation of methods and frameworks for evaluating quality issues in the MDE field is a challenge. The common purpose of these frameworks for the evaluation of quality is not enough to be able to compare these frameworks due to the diversity of concepts about the term *quality* that is applied in MDE projects. Validation procedures that use individual applications of quality frameworks do not allow the results to be generalized over the wide scope of the model-driven paradigm.

In this paper, we have reported the design and execution of a validation process for the MMQEF method through controlled experiments with fifty professionals in software engineering. The qualitative results from these experiments demonstrate the feasibility of the application and the use of the method by potential model-driven practitioners. However, the quantitative results also indicate the need to reinforce the current documentation of MMQEF in order to improve the deduction of quality inferences. Approaches from qualitative research were used to analyse the opinions and comments that were delivered by the participants in order to find evidence about the applicability of MMQEF and problems with the other quality methods used.

6.2 Impact

There are open challenges for the validation of MMQEF and other quality methods. The most relevant challenge is the application to software and system projects that are developed under the model-driven paradigm. The evidence that is presented in Section 4.1 demonstrates a clear influence of technical concerns for using artifacts of MDE. The evaluation of the applicability of quality methods such as MMQEF is highly dependent on the conviction about the central role of models in the development of complex systems and software projects.

MMQEF is not a revolutionary approach for evaluating quality in MDE. Instead, it can complement existing efforts to consolidate quality evaluation procedures by taking advantage of taxonomic analysis with a reference architecture for information systems (IS). The results that were obtained in the experiments preliminarily reflect the feasibility of the application of MMQEF. Because of the taxonomic structure of the reference architecture that is used in MMQEF, we think it is possible to harmonize the application of existing and new modelling languages and approaches based on their explicit association to the abstraction levels defined in model-driven architecture (MDA) specification and the concerns associated to information systems that are generally expressed as viewpoints.

The evaluation of quality issues in model-driven artifacts from an IS perspective could contribute to the adoption of MDE by explicitly managing the scope of the modelling artifacts regarding the IS concerns (which vary from organizational to technical levels) and by identifying the information that satisfies the relevant viewpoints in an IS.

The emphasis on the use of an IS reference architecture and its associated taxonomic structure makes it possible for MMQEF to be used with other quality initiatives for MDE by complementing and supporting specific quality dimensions that are related to IS concerns, such as semantics, pragmatics, and organizational (deontic) dimensions.

Therefore, to correctly address the application of quality methods for MDE such as MMQEF, more MDE scenarios are required, including roles that consider the use of models for critical decisions in a project (e.g. models to support architectural decisions). This requires more availability of technical and personal resources and time. The length of specific sessions such as that used in the experiment with MMQEF (3 hours) may be too short to demonstrate the impact of quality initiatives in real scenarios of practice.

6.3 Future work

In accordance with the *impact* stated above and taking into account the challenges for MMQEF, we have identified some further empirical evaluations that should be performed:

- Identify and evaluate the applicability, performance, and obtained quality of MDE scenarios in which MMQEF can be applied in order to evaluate and improve their quality.
- Identify the correspondence and potential integration of quality methods for MDE through comparisons of their key concepts and procedures for the identification and evaluation of quality.
- Demonstrate how MMQEF meets the principles of the MDE paradigm in scenarios of information systems development and software engineering projects.
- Improve the interaction with the taxonomy and activities that are proposed by MMQEF.
- Characterize the variables that allow the performance of methods for quality evaluation of MDE projects to be measured and compared.

Due to the resources that these activities require, we will consider the design and development of these works in the form of *case studies* or *action-research* techniques (Siau & Rossi, 1998).

7 Raw data

The supporting material and evidence of the performed experiment (i.e. the slides of the seminar, the questionnaires (forms) given to the participants, the obtained raw data, and results) can be found at https://github.com/fdgiraldo/MMQEFVAL/archive/master.zip.

References

Alaca, O. F., Tezel, B. T., Challenger, M., Goulo, M., Amaral, V., & Kardas, G. (2021). Agentdsm-eval: A framework for the evaluation of domain-specific modeling languages for multi-agent systems. *Computer Standards & Interfaces, 76*, 103513.

Arslan, S., & Kardas, G. (2020). Dsml4dt: A domain-specific modeling language for device tree software. *Computers in Industry, 115,* 103179.

Asici, T. Z., Tezel, B. T., & Kardas, G. (2021). On the use of the analytic hierarchy process in the evaluation of domain-specific modeling languages for multi-agent systems. *Journal of Computer Languages, 62,* 101020.

Bézivin, J. (2005). On the unification power of models. *Software & Systems Modeling, 4*(2), 171–188.

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

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

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

Fischer, M., & Strecker, S. (2018). A bibliography on: Evaluating conceptual models and modeling languages the quest for sensible evaluation criteria and methodic guidance (google docs). https://cutt.ly/Hf5gGTP

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

Giraldo, F. D., España, S., Pastor, O., & Giraldo, W. J. (2018). Considerations about quality in model-driven engineering. *Software Quality Journal, 26*(2), 685–750.

Giraldo, F. D., España, S., Pastor, O., & Giraldo, W. J. (2018). Evaluating the quality of a set of modelling languages used in combination: A method and a tool. *Information Systems, 77,* 48–70.

Giraldo, F. D., España, S., Giraldo, W. J., Pastor, Ó., & Krogstie, J. (2019). A method to evaluate quality of modelling languages based on the Zachman reference taxonomy. *Software Quality Journal, 27*(3), 1239–1269.

Gregor, S. (2006). The nature of theory in information systems. *MIS Q., 30*(3), 611–642.

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

Heggset, M., Krogstie, J., & Wesenberg, H. (2015). The influence of syntactic quality on pragmatic quality of enterprise process models. *Complex Systems Informatics and Modeling Quarterly Journal (CSIMQ), 5,* 1–13.

Hindawi, M., Morel, L., Aubry, R., & Sourrouille, J. L. (2009). Description and implementation of a uml style guide. In M. R. V. Chaudron, editor, *Mo-dels in Software Engineering*, pages 291–302, Berlin, Heidelberg, Springer Berlin Heidelberg.

ISO/IEC/(IEEE). ISO/IEC 42010:2011 : Systems and software engineering Architecture description, 12 2011.

Jedlitschka, A., Ciolkowski, M., & Pfahl, D. (2008). *Reporting Experiments in Software Engineering*, pages 201–228. Springer London, London.

Khalajzadeh, H., Simmons, A. J., Abdelrazek, M., Grundy, J., Hosking, J., & He, Q. (2020). An end-to-end model-based approach to support big data analytics development. *Journal of Computer Languages, 58,* 100964.

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

Krogstie J. (2012). *Model-Based Development and Evolution of Information Systems: A Quality Approach*, chapter Quality of Models, pages 205–247. Springer London, London.

Krogstie, J. (2012). *Model-Based Development and Evolution of Information Systems: A Quality Approach*, chapter Quality of Modelling Languages, pages 249–280. Springer London, London.

Lange, C., & 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.

Le Pallec, X., & 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. ACM.

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

Maes, A., & Poels, G. (2007). Evaluating quality of conceptual modelling scripts based on user perceptions. *Data & Knowledge Engineering, 63*(3), 701–724.

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

Mernik, M., Heering, J., & Sloane, A. M. (2005). When and how to develop domain-specific languages. *ACM Computing Surveys, 37*(4), 316–344.

Miranda, T., Challenger, M., Tezel, B. T., Alaca, O. F., Barišić, A., Amaral, V., et al. (2019). Improving the usability of a mas dsml. In D. Weyns, V. Mascardi, & A. Ricci (Eds.), *Engineering Multi-Agent Systems* (pp. 55–75)., pp Cham: Springer International Publishing.

Mohagheghi, P., Dehlen, V., & Neple, T. (2009). Definitions and approaches to model quality in model-based software development a review of literature. *Information and Software Technology*, 51(12):1646 – 1669. Quality of UML Models.

Moody, D. (2009). The "physics" of notations: Toward a scientific basis for constructing visual notations in software engineering. *IEEE Trans. Softw. Eng. 35*(6), 756–779.

Moody, D. L. (2003). The method evaluation model: a theoretical model for validating information systems design methods. In *Proceedings of the 11th European Conference on Information Systems, ECIS 2003, Naples, Italy 16-21 June 2003*, pages 1327–1336.

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

Santos, F., Nunes, I., & Bazzan, A. L. (2020). Quantitatively assessing the benefits of model-driven development in agent-based modeling and simulation. *Simulation Modelling Practice and Theory, 104,* 102126.

Shin, S. S. (2019). Empirical study on the effectiveness and efficiency of model-driven architecture techniques. *Software & Systems Modeling, 18*(5), 3083–3096.

Siau, K., & Rossi, M. (1998). Evaluation of information modeling methods-a review. In *System Sciences, 1998., Proceedings of the Thirty-First Hawaii International Conference on*, volume 5, pages 314–322 vol. 5.

Sowa, J. F., & Zachman, J. A. (1992). Extending and formalizing the framework for information systems architecture. *IBM Systems Journal, 31*(3), 590–616.

Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., & Regnell, B. (2012). *Experimentation in Software Engineering*. Springer.

Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B., & Wesslén, A. (2012). *Operation*, pages 117–122. Springer Berlin Heidelberg, Berlin, Heidelberg.

Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B., & Wesslén, A. (2012). *Planning*, pages 89–116. Springer Berlin Heidelberg, Berlin, Heidelberg.

Wortmann, A., Barais, O., Combemale, B., & Wimmer, M. (2019). Modeling languages in industry 4.0: an extended systematic mapping study. *Software and Systems Modeling*.

Zachman, J. A. (1987). A framework for information systems architecture. *IBM Systems Journal, 26*(3), 276–292.

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. 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 - now Ministry of Sciences) and is currently working with the PROS Research Center. He is a full associate 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 Senior Researcher by the Ministry of Sciences in Colombia. His research interests include software engineering, model-driven engineering, software quality, quality in model-driven engineering, software architecture, enterprise architecture and HCI, ORCID ID: https://orcid.org/0000-0002-6111-3055

Ángela J. Chicaiza is a Civil Engineer from the University of Quindío. Her research interests include software engineering, environment modelling, climate change, climate variability, data analytics and neuronal networks.

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 Politécnica de Valéncia (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 Politécnica de Valéncia (Spain). He received his Ph.D. in 1992. He was a researcher a 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.

Authors and Affiliations

Fáber D. Giraldo[1,3] · **Ángela J. Chicaiza**[1] · **Sergio España**[2] · **Óscar Pastor**[3]

Fáber D. Giraldo
fdgiraldo@uniquindio.edu.co; fdgiraldo@pros.upv.es

Sergio España
s.espana@uu.nl

Óscar Pastor
opastor@pros.upv.es

[1] SINFOCI Research Group, University of Quindío, Cra 15 Calle 12N, Armenia Quindio, 630004 Colombia, USA

[2] Department of Information and Computing Sciences, Utrecht University, Office: Buys-Ballotgebouw (BBL) 580, Utrecht 3508 TB, P.O. Box 80.089, Netherlands

[3] PROS Research Centre, Universitat Politècnica de València, Camino de Vera S/N Valencia, 46022 Velencia, Spain