Evaluation of Requirements Models

Catarina Gralha
NOVA Laboratory for Computer Science and Informatics
Department of Computer Science
Faculdade de Ciências e Tecnologia
Universidade NOVA de Lisboa
Website: http://microlina.github.com
Email: acg.almeida@campus.fct.unl.pt

***Abstract*—Requirements Engineering (RE) approaches, following paradigms such as goal-oriented [1] or scenario-based [2], provide expressive model elements for requirements elicitation and analysis. However, these approaches are still struggling when it comes to managing the quality of their models. Problems in quality can cause difficulties in managing and understanding requirements, which in turn leads to increased development costs. The models' quality should then be a permanent concern. We propose a quantitative assessment of the goal-oriented and scenario-based models' quality, namely its complexity, completeness, appropriateness recognizability, understandability and learnability. To this end, we propose a combination of techniques to be applied to the RE models, the modelling process, and the models' notation. We are going to define metrics about the models, through the Goal-Question-Metric (GQM) approach [3], and incorporate them in a common evaluation framework that helps in the requirements modelling process. The quality of the RE models, the modelling process and the model's notation will be measured by collecting biometric data from stakeholders, by using eye-tracking devices, electroencephalography (EEG) scanners, and electro-dermal activity (EDA) scanners. Furthermore, we will collect metrics about the model during the modelling process, and the subjective opinion of stakeholders about the usage of these models, through questionnaires like NASA TLX [4] (which measures perceived effort while working on tasks). All metrics and biometrics are going to be theoretically and experimentally evaluated, through a set of case studies and experiments with different types of participants (including researchers, practitioners and students).**

***Index Terms*—requirements models, quality evaluation, metrics, biometrics**

I. Introduction

Requirements models are often used for requirements elicitation and analysis, where communication with different types of stakeholders plays a major role. For this communication to be effective, both requirements engineers and other stakeholders need to have a common understanding of the requirements models [5]. However, as a common challenge, requirements engineering approaches are still struggling when it comes to managing the quality of their models, and problems in quality can cause difficulties in the management and understanding of those models, leading to increased development costs. These difficulties in understanding the model can introduce validation errors: a stakeholder may not correctly understand a given model (due to its accidental complexity [6], for example) and accept a specification that does not meet his needs. Even when the model is accepted, other problems in quality, such as incomplete or unnecessarily complex specifications, may jeopardise the correct implementation of the software system. Since the cost of repairing an error made in the requirements elicitation phase increases along the next phases of a software project [7], it is imperative that problems are detected as soon as possible. Therefore, the quality of the models should be a permanent concern.

Quality attributes such as complexity, completeness, appropriateness recognizability, understandability, and learnability should be measured and monitored during and after the requirements modelling activity, so that corrective actions can be undertaken in a timely manner, saving important resources. Measuring quality attributes while the models are being built can give us insights about how the model is created, and what is the actual effort required for both its creation and modification. Additionally, measuring quality attributes is also important when models are fully finished. Post-mortem analysis can support an evidence-based understanding on how the modelling language constructs are used, in practice. Furthermore, it can provide useful information on what is the actual relationship between different types of stakeholders and the model, in terms or their ability and effort to understand and review it, and by providing data on which specific parts of the models are more problematic. Conducting these analysis, in both successful and unsuccessful projects, helps to highlight strengths and opportunities for models improvement. With a quantitative assessment of the requirements models' quality, it is possible to promote adjustments and changes in the development process, in order to reduce or eliminate the causes of problems that notably affect the production of a given software system. In the end, by identifying quality problems, it is possible to characterise and analyse them to look for patterns of wrong usage or understanding of the modelling language. This type of information can provide useful insights for the evolution process of the modelling approaches themselves.

A. Main Goals

The main objective of this thesis is to evaluate the quality of goal-oriented and scenario-based models, in terms of its complexity, completeness, appropriateness recognizability, understandability, and learnability. By understanding the quality problems that affect those models, it is possible to identify opportunities for their improvement. Accordingly, the

approach is aimed at providing an integrated framework for the evaluation of these models, to support on-the-fly warnings (while creating and editing those models) about potential quality problems and advice on how to mitigate them. In the end, this will help in the stakeholders engagement and empowerment in the requirements engineering process, by improving the communication and involvement between them (both IT and non-IT professionals).

There are several ways to evaluate the complexity and the completeness of a model, one of them being the collection of metrics about it, such as the percentage of incoming and outgoing relationships of a given model element, or the specific level of detail of that element, for example. This analysis is useful for understanding the model as a whole, but collecting product metrics may not be enough since it does not give us insights about the relationship between stakeholders and the model. To do so, we need to measure the success on tasks such as understanding, reviewing and modifying models, by collecting (*i*) direct task performance metrics such as precision, recall, F-measure, and duration of those tasks; or (*ii*) indirect measures such as the visual effort while performing them (assessed with eye-tracking devices [8]), and the participants perceptions on their effort while performing the tasks, measured, for example, with a NASA Task Load Index (NASA TLX) questionnaire [4]. Although eye-tracking devices can give insights into where a subject is directing his eyes at a given time and how eye-movements are modulated by visual attention, tracking gaze positions alone does not inform about cognitive processes and the emotional states that guided the eye-movements. In these cases, eye-trackers can be complemented by other biometric sensors, such as electroencephalography (EEG), which can be used to measure mental effort, in terms of concentration, por example, while a participant is performing a given task [9]; and electro-dermal activity (EDA) scanners, which can be used to measure stress [10]. By using these equipments, we can capture a broader view of the human behaviour in that particular moment, gaining meaningful insights into the dynamics of attention, motivations, and emotion.

The combination of all these techniques will give us a full picture of the models, the problems they may have, and the way stakeholders interact with them. By combining these techniques, we can say, for example, that a given model is complex because it has x elements, which surpasses a threshold that must not be exceeded for a better understanding of the model. The identification of that threshold is possible by analysing how participants react (through the usage of biometric sensors) to models of different sizes, by analysing their success on model construction, modification, understanding and reviewing tasks, and by studying their own perception on success, effort and difficulty while performing those tasks.

B. Contributions

The results of this thesis will contribute to software development in general, and requirements engineering in particular, with a framework for improving the requirements definition process with an early detection of quality problems in the requirements models. It will also contribute to stakeholders empowerment. In particular, the expected contributions are: (*i*) define a generic approach for the quality evaluation of goal-oriented and scenario-based models, in terms of complexity, completeness, appropriateness recognizability, understandability and learnability; (*ii*) generalise an initial measurement tool [11], [12], which currently supports the evaluation of the complexity, completeness and correctness of *i** goal models, to other GORE and scenario-based approaches; (*iii*) extend our metrics set to cover other quality attributes: appropriateness recognizability, understandability and learnability; (*iv*) evaluate the proposed metrics by applying them to a group of case studies; (*v*) evaluate the usability of the requirements notation (in terms of appropriateness recognizability, understandability and learnability), from the perspective of ordinary users and requirements engineers, by using biometric equipment like electroencephalograms recording machines and eye-trackers, combined with process metrics such as effort (measured in terms of time to complete requirements engineering activities: construction, modification, understanding and reviewing), as well as of the achieved correctness in those activities.

II. State of the Art

Quality assessment of conceptual models has been studied several times over the years, and different frameworks to perform that evaluation have been proposed, such as the ones of Lindland et al. [13] and Krogstie et al. [14]. Indeed, due to the number of different proposals, none of them widely accepted in practice, Moodie et al. [15] felt that there was a need to have a common evaluation framework, since proliferation of quality frameworks is counterproductive.

Mayerhofer [16] states that methods for ensuring the quality of models can be divided into two fields: static analysis of models, and dynamic analysis of models. According to her, static analysis methods verify the correctness of models by assessing its static properties, whereas dynamic analysis methods verify the quality of models by executing them. Also, the currently available tool support for model testing and debugging is still insufficient. Guizzardi et al. [17] state that an approach to conceptual modelling requires tools for modellers to gain confidence on the quality of the models they produce, and to be able to develop high-quality models, a modeller must have the support of expressive engineering tools. They proposed a tool that is able to automatically identify anti-patterns in user's models, provide visualisation for its consequences, and generate corrections to these models by the automatic inclusion of OCL constraints.

Ramos et al. [18] claim that early identification of syntactic problems (e.g., large and unclear descriptions, duplicated information) and the removal of their causes, can improve the quality of use case models. They describe the AIRDoc approach, which aims to facilitate the identification of potential problems in requirements documents using refactoring and patterns. To evaluate use case models, the AIRDoc process uses the GQM approach.

According to Monperrus et al. [19], metrics are a practical approach to evaluate properties of domain-specific models, but it is costly to develop measurement software for each one of them. They present a model-driven and generative approach to measuring models, that is domain-independent. Furthermore, several studies have been carried out regarding the quality evaluation of requirements models, by using metrics. In this field, some of the most studied quality attributes are understandability and comprehensibility, efficiency, correctness, defect rate, completeness and consistency, confinement and changeability [20], [21]. However, the majority of the studies are related with the quality evaluation of UML models [21].

Regarding the evaluation of other requirements models, Espada et al. [22] proposed and validated a metrics suite for evaluating completeness and complexity of KAOS goal models, formally specified (using OCL) and incorporated in a KAOS modelling tool. The metrics suite was evaluated with several real world case studies. In previous work, we have defined, implemented, and validated complexity and completeness metrics for *i** models [11], [12]. In terms of complexity, we have identified refactoring opportunities to improve the modularity of *i** models, and consequently reduce their complexity. Regarding completeness, we were able to automatically detect model incompleteness, helping requirements engineers to evaluate how close they are to completing their models. In this thesis, we plan to extend this work to other goal-oriented and scenario-based models.

In this thesis, we are also interested in the impact of model quality on different activities performed by stakeholders upon those models. That impact can be studied in several ways, one of them being through biometrics. Most of the studies in software engineering using biometrics have focused on eye-tracking technology and have investigated how developers comprehend code (see, for example, [23], [24], [25]). More recently, eye-tracking has been used to assess the effort involved in the comprehension of software models, by monitoring participant's visual attention through fixations and other indicators [8]. Moody [26] and Caire et al. [5] propose approaches to help improving the understandability of requirements models, by improving the concrete syntax of those models through the definition of a set of principles for designing cognitively effective visual notations. Yusuf et al. [27] used eye-tracking to compare the visual effort involved in answering questions about UML class diagrams containing the same information, but designed following 3 different layout strategies. Fritz et al. [9] and Störrle et al. [28] propose approaches to classify the difficulty of code or models comprehension, respectively, by using biometric data collected via electroencephalographic activities and eye-tracking. Only very few studies investigated the use of other biometric sensors rather than eye-trackers. Siegmund et al. [29] examined the active brain regions during small code comprehension tasks using fMRI technology. Parnin et al. [30] used electromyography to measure developers' sub-vocal utterances and found that these utterances might be used to measure programming task difficulty.

Although some work has been performed regarding the evaluation of requirements models, the combinations of product metrics, different types of biometrics, success rate of the performed tasks, and the stakeholders perceptions (subjective opinion) on their success and effort has not yet been explored, which we plan to do in this thesis. Furthermore, the use of biometrics to evaluate the quality of requirements models is still in the early stages.

III. Proposed Approach

In this work, we are interested in the quality evaluation of goal-oriented and scenario-based requirements models, by using more traditional metrics and biometrics sensors. We have defined three hypotheses for our research. In order to be able to evaluate those hypotheses, we have defined a series of research questions.

- **Hypothesis 1.** *Product metrics are an efficient and pragmatic way to assess and measure the quality of requirements models.*
  - **RQ 1:** *To what extent can product metrics help in evaluating the quality of a requirements model?*
- **Hypothesis 2.** *Affordable (low cost) biometrics are a reliable way to measure the difficulty a stakeholder experiences while working on requirements models' construction, modification, understanding and reviewing tasks.*
  - **RQ 2:** *How can we use biometric sensors to capture a stakeholders' perceived difficulty while working on a task?*
  - **RQ 3:** *How can we use biometric measurements to accurately predict whether tasks are difficult or easy to perform by a given stakeholder?*
- **Hypothesis 3.** *A model with a higher quality level improves the performance of stakeholders during modification, understanding and reviewing tasks on requirements models.*
  - **RQ 4:** *What is the relationship between the quality level of a requirements model and the ability of a stakeholder to modify, understand, or review it?*

To address the identified objectives and hypotheses, the approach consists on four main phases: systematic literature review, planning and design, implementation, and evaluation, which are described next.

A. Systematic Literature Review

A systematic literature review was already carried out on the usability of requirements techniques (see section IV-A for further details). A systematic literature review will be conducted, by following Kitchenham and Charters' guidelines [31], regarding the usage of biometric equipment in software engineering. Our main goal is to identify specific equipments and techniques that are being explored, more particularly the ones that are being used for quality evaluation of requirements models. Sharafi et al. [8] performed a systematic literature

review about the usage of eye-tracking devices in software engineering, which did not include other biometrics equipments (such as EEG and EDA scanners).

B. Planning and Design

This phase is concerned with (*i*) the quality attributes of the requirements models and the definition of metrics to support their evaluation, by following the Goal-Question-Metric (GQM) approach [3], and (*ii*) the design of the experiments conducted and to be conducted with different types of stakeholders.

First and foremost, it is necessary to select the quality attributes that are going to be considered for models' evaluation, to fuel the GQM process. We have selected 5 quality attributes: complexity, completeness, appropriateness recognizability, understandability and learnability. Each one of these can be characterised as a **goal**:

- **Goal 1:** Evaluate the complexity of goal-oriented and scenario-based models
- **Goal 2:** Evaluate the completeness of goal-oriented and and scenario-based models
- **Goal 3**: Evaluate the appropriateness recognizability of goal-oriented and and scenario-based models
- **Goal 4:** Evaluate the understandabily of goal-oriented and and scenario-based models
- **Goal 5:** Evaluate the learnability of goal-oriented and and scenario-based models

For goals 1 and 2 we will start by using product metrics, followed by biometrics. For the remaining goals we will also use biometrics combined with success, time and perceived difficulty. Each goal is refined into several **questions** that usually break down the issue into its major components, characterising how a given goal can be achieved. We need to define particular questions for each one these goals. For example, for goal 1 and 2, we may have the (corresponding) following questions:

- **Question 1:** How dependent is a model element, with regard to its outgoing relationships?
- **Question 2:** How close are we to finish the relationships between different model elements?

Each question is then refined into **metrics**, which provide quantifiable information needed to answer those questions.

Regarding eye-trackers (used for goals 3, 4 and 5), we have already selected some of the metrics:

- **Fixation rate on relevant elements:** the fraction of number of fixations in an given AOI (Area of Interest) over the total number of fixations in the AOG (Area of Glance).
- **Fixation rate on irrelevant elements:** the fraction of number of fixations in an given AOI over the total number of fixations in the AOG.
- **Average duration of relevant fixation:** the fraction of total duration of fixations for relevant AOIs over the number of elements of the relevant AOIs.
- **Average duration of irrelevant fixation:** the fraction of total duration of fixations for relevant AOIs over the number of elements of the relevant AOIs.

The next step is the formalisation of the product metrics, by designing heuristics defined in OCL [32], so that they can be easily integrated in our framework (see Implementation III-C).

Regarding the experiments with stakeholders, where biometrics are going to be used, we need to plan it in a systematic way, starting by the definition of hypothesis (e.g. biometric equipment are a reliable way to measure the effort to understand and review requirements models), research questions, and protocol to be followed. We need to conduct different experiments for goals 3, 4 and 5. We have already conducted one quasi-experiment on the impact of model layouts on the effort required for understanding and reviewing *i** Strategic Rationale models (see section IV-B for further details).

C. Implementation

This phase is concerned with the implementation of a framework that allows not only the creation of requirements models, but also the automated collection of product metrics about those models. Since manually collecting the metrics is time-consuming and error-prone, having a tool that collects this information is essential. An initial measurement tool, which currently supports the complexity, completeness and correctness of *i** models, was already developed [11], [12], using Domain Specific Languages construction mechanisms and tools. The same mechanisms will be used to build the new framework, which will support other goal-oriented and scenario-based models. The implementation of metrics as being part of the tool will also be carried out, so that they can be automatically evaluated on the requirements models built with the framework.

D. Evaluation

This phase is concerned with the evaluation of the proposed metrics, the underlying tool-set, and the RE approach itself. For the evaluation of the proposed metrics, one needs to define a quality model for the evaluation of the process of using metrics-based approaches to detect RE models quality improvement opportunities. The next step is to identify and select case studies for the RE approaches and model them with the tool. The resulting models will be used to collect metrics values about them. During the modelling process, with support provided by the modelling tool itself, we can collect metrics values about the models and be able to identify opportunities for their improvement/refactoring, through the information given by experiments with stakeholders. Another task is to conduct a post-mortem analysis of the models, to learn about the modelling trends using RE approaches.

For the evaluation of the RE approach itself, one needs to evaluate the usability (in terms of appropriateness recognizability, understandability and learnability) of the notations of the RE approaches from the perspective of ordinary users, by using biometric equipment like electroencephalograms recording machines, eye-tracking devices and other cognitive

processes. These techniques will also be used when analysing quality attributes, such as complexity, and will be useful to assess the effort spent by the user in order to create, change, understand and review different models. We will follow the process described in Jedlitschka et al. [33].

In practice, we plan to conduct an experiment with different types of participants, for each one of the quality attributes that we want to evaluate. In that sense, we will have an experiment, for instance, for evaluating learnability. In this experiment, participants with no prior experience with a given modelling languages will be asked, after having learning sessions, to model a given set of requirements. For appropriateness recognizability, for example, a group of participants will define a set of requirements, and analyse if and which one of a given collection of models is appropriate for their previously defined requirements. During all these experiments, we will collect information from eye-trackers, EEG and EDA scanners, the success rate of the performed tasks, and the participants perceptions on their effort.

IV. Current State

A. Systematic Literature Review

We have performed a systematic literature review on the usability of requirements techniques [34], since it has been recognised as a key factor for their successful adoption by industry. RE techniques must be accessible to stakeholders with different backgrounds, so they can be empowered to effectively and efficiently contribute to building successful systems. When selecting an appropriate requirements engineering technique for a given context, one should consider the usability supported by each of the candidate techniques. The first step towards achieving this goal is to gather the best evidence available on the usability of RE approaches by performing a systematic literature review. We answer the following research question: *How is the usability of requirements engineering techniques and tools addressed?*

We systematically review articles published in the Requirements Engineering Journal, one of the main sources for mature work in RE, to motivate a research roadmap to make RE approaches more accessible to stakeholders with different backgrounds. In the future, we plan to replicate this SLR to include other venues. The search on this journal database resulted in over 400 papers, of which over 60 were selected in a first iteration of the process (based on automatic search). From those, 35 remained for extraction, after screening the titles and abstracts. Of these, 19 were selected for data extraction and further analysis.

We observed that there is relatively little evidence concerning the usability of the requirements engineering approaches, denoting this has not been a top priority concern in the past. That said, we found a large variety of approaches submitted to some form of usability assessment, so it is fair to say the RE community is increasingly concerned about the problem of making its approaches usable not only for requirements engineers, but also to stakeholders, with their diverse backgrounds and needs. We expect to find an increasing number of studies concerned with usability in the near future, consistently with what we are observing in other software engineering communities. Although validations with students and academic examples are still the most frequent kind of evaluations reported, the RE community is pushing for evaluations with professional practitioners, in industrial settings, to increase the results validity and its applicability to real work environments.

B. Quasi-experiment

We have conducted a quasi-experiment to assess the impact of model layouts, by triangulating the success level in understanding and reviewing tasks, the required effort to accomplish them, and eye-tracking information monitoring how stakeholders explore *i** diagrams during their tasks [35].

Our goal was to evaluate the effect of the layout guidelines on the *i** novice stakeholders' ability to understand and review those models. In this quasi-experiment, participants were given two understanding and two reviewing tasks. Both tasks involved a model with a bad layout and another model following the *i** layout guidelines. We evaluated the impact of layouts by combining the success level in those tasks and the required effort to accomplish them. Effort was assessed using time, perceived complexity (with NASA TLX [4]), and eye-tracking data.

We concluded that participants were more successful in understanding than in reviewing tasks. However, we found no statistically significant difference in the success, time taken, or perceived complexity, between tasks conducted with models with a bad layout and models with a good layout. Most participants had little to no prior knowledge in *i**, making them more representative of stakeholders with no requirements engineering expertise. They were able to understand the models fairly well after a short video tutorial, but struggled when reviewing models. Adherence to the existing *i** layout guidelines did not significantly impact *i** model understanding and reviewing performance, at least for diagrams of this size and complexity (2 actors and $\approx$ 20 elements). However, we expect layout quality to have a stronger impact as diagrams increase in size and complexity, in line with findings on models expressed with other languages (e.g. with UML).

At the moment, a replication of this experience is also being performed in another university and we plan to replicate it again in other institutional contexts (i.e., industry). We also plan to explore the impact of alternative concrete syntaxes in the understandability of requirements models. Although these early evaluations are conducted with some models like *i** (a goal-oriented model), the evaluation approach itself is generic and applicable to other requirements languages.

C. Accepted publications

In previous work, we have defined, implemented, and validated complexity and completeness metrics for *i** models, which were published in CAiSE 2014 [11] and in the Information Systems journal [12]. The systematic literature review

mentioned in section IV-A was published in ACM SAC 2016 [34]. The results of the quasi-experiment mentioned in section IV-B was accepted for publication in RE 2016 [35].

V. Conclusions

In this Ph.D thesis, we propose an approach to support the quantitative assessment of goal-oriented and scenario-based models' quality. This approach will also allow us to assess the usability (in terms of appropriateness recognizability, understandability and learnability) of different requirements modelling approaches. We plan to follow the phases previously described in section III, extending a previous analysis to other requirements models and languages, with a broader set of quality attributes, in order to identify opportunities for their improvement, promoting adjustments and changes in the development process. To this end, we will use both metrics and biometrics, combining them to have a full picture about the model, and the relationship between the model and different types of stakeholders.

Acknowledgment

This work was partially funded by NOVA LINCS UID/CEC/04516/2013 and FCT-MCTES in the context of the research grant SFRH/BD/108492/2015.

References

[1] A. Van Lamsweerde, "Goal-oriented requirements engineering: A guided tour," in *Proceedings of the fifth IEEE International Symposium on Requirements Engineering, 2001*. IEEE, 2001, pp. 249–262.

[2] I. F. Alexander and N. Maiden, *Scenarios, stories, use cases: through the systems development life-cycle*. John Wiley & Sons, 2005.

[3] V. Basili, G. Caldiera, and H. D. Rombach, *Goal Question Metric (GQM) Approach*, 1st ed. John Wiley & Sons, Inc., 1994.

[4] S. G. Hart and L. E. Staveland, "Development of nasa-tlx (task load index): Results of empirical and theoretical research," *Advances in psychology*, vol. 52, pp. 139–183, 1988.

[5] P. Caire, N. Genon, P. Heymans, and D. L. Moody, "Visual notation design 2.0: Towards user comprehensible requirements engineering notations," in *Requirements Engineering Conference (RE), 2013 21st IEEE International*. IEEE, 2013, pp. 115–124.

[6] F. P. Brooks, *The Mythical Man-Month: Essays on Software Engineering*. USA: Addison-Wesley, 1995.

[7] A. M. Davis, *Software Requirements: Objects, Functions, and States*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1993.

[8] Z. Sharafi, Z. Soh, and Y.-G. Guéhéneuc, "A systematic literature review on the usage of eye-tracking in software engineering," *Information and Software Technology*, vol. 67, pp. 79–107, 2015.

[9] T. Fritz, A. Begel, S. C. Müller, S. Yigit-Elliott, and M. Züger, "Using psycho-physiological measures to assess task difficulty in software development," in *Proceedings of the 36th International Conference on Software Engineering*. ACM, 2014, pp. 402–413.

[10] S. C. Müller and T. Fritz, "Stuck and frustrated or in flow and happy: Sensing developers' emotions and progress," in *Proceedings of the 37th International Conference on Software Engineering-Volume 1*. IEEE Press, 2015, pp. 688–699.

[11] C. Gralha, M. Goulão, and J. Araújo, "Identifying modularity improvement opportunities in goal-oriented requirements models," in *Advanced Information Systems Engineering*. Springer, 2014, pp. 91–104.

[12] C. Gralha, J. Araújo, and M. Goulão, "Metrics for measuring complexity and completeness for social goal models," *Information Systems*, vol. 53, pp. 346–362, 2015.

[13] O. I. Lindland, G. Sindre, and A. Solvberg, "Understanding quality in conceptual modeling," *Software, IEEE*, vol. 11, no. 2, pp. 42–49, 1994.

[14] J. Krogstie, G. Sindre, and H. Jørgensen, "Process models representing knowledge for action: a revised quality framework," *European Journal of Information Systems*, vol. 15, no. 1, pp. 91–102, 2006.

[15] D. L. Moody, "Theoretical and practical issues in evaluating the quality of conceptual models: current state and future directions," *Data & Knowledge Engineering*, vol. 55, no. 3, pp. 243–276, 2005.

[16] T. Mayerhofer, "Testing and debugging uml models based on fuml," in *Software Engineering (ICSE), 2012 34th International Conference on*. IEEE, 2012, pp. 1579–1582.

[17] G. Guizzardi and T. P. Sales, "Detection, simulation and elimination of semantic anti-patterns in ontology-driven conceptual models," in *Conceptual Modeling*. Springer, 2014, pp. 363–376.

[18] R. Ramos, J. Castro, J. Araújo, A. Moreira, F. Alencar, E. Santos, R. Penteado, S. Carlos, and S. Paulo, "Airdoc–an approach to improve requirements documents," in *22th Brazilian Symposium on Software Engineering (SBES'08)*, 2008.

[19] M. Monperrus, J.-M. Jézéquel, B. Baudry, J. Champeau, and B. Hoeltzener, "Model-driven generative development of measurement software," *Software & Systems Modeling*, vol. 10, no. 4, pp. 537–552, 2011.

[20] N. Condori-Fernandez, M. Daneva, K. Sikkel, R. Wieringa, O. Dieste, and O. Pastor, "A systematic mapping study on empirical evaluation of software requirements specifications techniques," in *Proceedings of the 2009 3rd International Symposium on Empirical Software Engineering and Measurement*. IEEE Computer Society, 2009, pp. 502–505.

[21] P. Mohagheghi, V. Dehlen, and T. Neple, "Definitions and approaches to model quality in model-based software development – a review of literature," *Information and Software Technology*, vol. 51, no. 12, pp. 1646 – 1669, 2009.

[22] P. Espada, M. Goulão, and J. Araújo, "A framework to evaluate complexity and completeness of kaos goal models," in *Advanced Information Systems Engineering*. Springer, 2013, pp. 562–577.

[23] M. E. Crosby and J. Stelovsky, "How do we read algorithms? a case study," *Computer*, vol. 23, no. 1, pp. 25–35, 1990.

[24] R. Bednarik and M. Tukiainen, "An eye-tracking methodology for characterizing program comprehension processes," in *Proceedings of the 2006 symposium on Eye tracking research & applications*. ACM, 2006, pp. 125–132.

[25] B. Sharif and J. I. Maletic, "An eye tracking study on camelcase and under_score identifier styles," in *Program Comprehension (ICPC), 2010 IEEE 18th International Conference on*. IEEE, 2010, pp. 196–205.

[26] D. L. Moody, "The "physics" of notations: toward a scientific basis for constructing visual notations in software engineering," *Software Engineering, IEEE Transactions on*, vol. 35, no. 6, pp. 756–779, 2009.

[27] S. Yusuf, H. Kagdi, J. Maletic *et al.*, "Assessing the comprehension of uml class diagrams via eye tracking," in *ICPC'07*. IEEE, 2007, pp. 113–122.

[28] H. Störrle, N. Baltsen, H. Christoffersen, and A. Maier, "On the impact of diagram layout: How are models actually read?" in *International Conference on Model Driven Engineering Languages and Systems (MoDELS) 2014*, 2014, pp. 31–35.

[29] J. Siegmund, C. Kästner, S. Apel, C. Parnin, A. Bethmann, T. Leich, G. Saake, and A. Brechmann, "Understanding understanding source code with functional magnetic resonance imaging," in *Proceedings of the 36th International Conference on Software Engineering*. ACM, 2014, pp. 378–389.

[30] C. Parnin, "Subvocalization-toward hearing the inner thoughts of developers," in *Program Comprehension (ICPC), 2011 IEEE 19th International Conference on*. IEEE, 2011, pp. 197–200.

[31] B. A. Kitchenham and S. Charters, "Guidelines for performing Systematic Literature Reviews in Software Engineering," p. 65, 2007.

[32] Object Management Group, "Object constraint language (ocl)," Last access May 2016. [Online]. Available: http://www.omg.org/spec/OCL/

[33] A. Jedlitschka, M. Ciolkowski, and D. Pfahl, "Reporting experiments in software engineering," in *Guide to advanced empirical software engineering*. Springer, 2008, pp. 201–228.

[34] D. Bombonatti, C. Gralha, A. Moreira, J. Araújo, and M. Goulão, "Usability of requirements techniques: A systematic literature review," *Proceedings of the 31st ACM Symposium on Applied Computing - Requirements Engineering Track (SAC 2016)*, 2016.

[35] M. Santos, C. Gralha, M. Goulão, J. Araújo, A. Moreira, and J. Cambeiro, "What is the impact of bad layout in the understandability of social goal models?" in *Requirements Engineering Conference (RE) (to appear)*, 2016.