Asian Journal of Multidisciplinary Studies
Vol. 1, No. 2, (2018)
ISSN 2651-6691 (Print)
ISSN 2651-6705 (Online)

Quality Assurance in UML Models: A Review of Related Literature

Ferdinand V. Dalisay
fvdalisay@psu.edu.ph
Pangasinan State University

Abstract *- The quality UML models directly affects the quality of the software or application being developed. Therefore, quality assurance in UML models should be regarded as an important niche. This paper describes a literature review of peer-reviewed journal articles published from 2012 through 2018 on the quality of conceptual models written in UML.*

Keywords: quality assurance, UML models

1 INTRODUCTION

As years pass by, software and applications are increasingly becoming complex. The complexity is very evident that it is impossible to test every aspect of the system software or application programs for release. Through the models, however, one can understand how the software works and the complexity of the software is reduced [1]. Over the years, we have come to understand that modeling offers benefits to different stakeholders in software projects [2]. In the early stages of application development, models aid in understanding and communicating the system requirements. During development, architecture and design models guide the implementation of the system [3]. Finally, models are used for test-generation [4] and for easing maintenance activities [5].

Agile-centric software development itself has become more model reliant. The OMG Model Driven Architecture [6] and the recent growth of the Model-Driven Development (MDD) software engineering paradigm [7] emphasizes the role of modeling in the development of software systems. MDD treats software development as a set of transformations between successive models from requirements to analysis, to design, to implementation, and to deployment. MDD's defining characteristic is that software development's primary focus and products are models rather than computer programs.

The Agile software development made a change in the dominant question [8] "Should we do modeling?" to "How should we do modeling?". The question denotes a new focus to the modeling process, rather than on the software product resulting from the development activities, and this puts model quality in the forefront. There has been increasing interest, both in industry and academia, on methods and techniques for quality assessment, assurance, and improvement of models in software development and maintenance [9]. While there has been a great deal of research on software quality, there has been relatively little work on the quality of models, and the concept of model quality is poorly understood. Existing knowledge on software quality has limited applicability to

model quality. Models have very different characteristics than source code because models have multiple views, may be used informally and casually rather than formally and with precision, can be used throughout all phases of the project, and so on. In order to advance the field of quality in software modeling, it is useful to explore its current state by locating, evaluating, and interpreting relevant research to date that is related to model quality with a focus on UML.

This paper presents a literature review of papers dealing with the quality of UML models. A proper literature review follows a rigorous and systematic approach.

This paper is structured as follows. The next section discusses the steps to be undertaken in the literature review. This includes the steps in general database search, filtering and analysis of the results. Section 3 discusses the results and interpretation of data. Lastly, section 4 concludes the paper.

2 METHODOLOGY

This paper uses a review of related literature to explore published journals and literatures regarding the quality of UML models. The following are the steps undertaken in doing the literature review:

2.1 General database search

Related keywords were used for a general database search. The main keywords are: "UML Modeling" and "quality". The keywords listed above were used in this academic database: ACM Digital Library. The search result from the general database search will be initially filtered by the publishing year. All journal articles and papers published from 2012 up to present will be included in the initial list of literatures to be reviewed

2.2 Filtering and Analysis of the Results

The initial list of literatures will be reviewed according to its relevance to the research topic. Reading all of the papers and research articles from the results of a general database search can be a very tedious task and may not be feasible because of time constraints. Thus, it was determined that reading the abstract and conclusion is a more realistic approach in filtering the results.

After reading the abstract and conclusion of the researches and literatures from the filtered list, it must also be evaluated according to its relevance to the main research question of this paper: What are the UML model quality used?

2.3 Categorization of Research Papers

The research papers resulted from filtering will be further categorized according to quality. There are three main model quality types: syntactic quality, semantic quality, and pragmatic quality [10]. Each of these quality types contains some additional quality characteristics. Most of these definitions are taken from (International Organization for Standardization, 1998) which is related to software product quality. The table below shows the characteristics per model quality type.

Table 1: Number of search results per query

Type of Quality	Characteristics
Syntactic quality	Correctness
Semantic quality	consistency, completeness, correctness
Pragmatic quality	maintainability, analyzability, understandability,

	testability, functionality, executability, reusability, complexity, dependability

3 RESULTS AND DISCUSSION

3.1 Results of general database search

Table 2 shows the initial search results for each of the three search queries formulated.

Table 2: Number of search results per query

Search Query	No. of Search Results (ACM Digital Library)
Quality of UML Modeling	**476,557**

To further narrow down the number of search results, it has been filtered by publication year. Table 3 shows the number of search results after filtering by publication date.

Table 3: Number of search results per query

Search Query	No. of Search Results (ACM Digital Library)
Quality of UML Modeling	**174,448**

3.2 Results of Filtering and Analysis of the Results

The number of search results from the filtering done above is still too many to manage. Therefore, additional filtering was done. The search query was modified to only include journal articles that matches all of the search keywords. Table 4 shows the number of search results after filtering by publishing date.

Table 4: Number of search results per query

Search Query	No. of Search Results (ACM Digital Library)
Quality of UML Modeling	**68**

Each of the paper's abstract and conclusion were read to ensure its relevance to the research question. The table below shows the pilot studies to be discussed.

Table 5: Pilot studies

No.	Paper Title
1	Dependability Modeling and Analysis of Software Systems Specified with UML [11]
2	The Use of UML Class Diagrams and Its Effect on Code Change-proneness [12]
3	SUDA: A Scenario-based UML Design Analysis Approach [13]
4	Verification and Validation of UML Conceptual Schemas with OCL Constraints [14]
5	Diagram Size vs. Layout Flaws: Understanding Quality Factors of UML Diagrams [15]
6	Towards a UML Profile for Data Intensive Applications [17]
7	DSL-based Support for Semi-Automated Architectural Component Model Abstraction Throughout the Software Lifecycle [19]

8	Combining fUML and Profiles for Non-Functional Analysis Based on Model Execution Traces [20]
9	Training of Requirements Analysis Modeling with UML-based Prototype Generation Tool [21]
10	UMLx: a UML diagram analytic tool for software management decisions [22]
11	Executing and Debugging UML Models: an fUML extension [23]

The table below shows the categorized journal articles according to quality approach.

Table 6: Categorized journal articles

Type of Quality	Journal Article
Syntactic quality	[14]
Semantic quality	[13][15][21][22]
Pragmatic quality	[11][12][17][19][20][23]

The table below shows the percentage of journal articles associated to quality approach.

Table 7: Percentage of journal articles associated to quality approach

Type of Quality	Journal Article
Syntactic quality	9.10%
Semantic quality	36.36%
Pragmatic quality	54.54%
TOTAL	100%

From the results above, only 1 paper out of the 11 pilot studies selected falls to the syntactic quality. Syntactic quality refers to how well the model adheres to the rules of the language. It is also known as syntactic correctness. The word correctness refers to the absence of syntactic errors, meaning that the model is a valid instantiation of the metamodel that defines the UML type of diagram considered. The paper of Queralt and Teniente [14] introduced a method to ensure the correctness of the UML diagram. They introduced an approach to help the designer in the verification and validationof a UML conceptual schema, with arbitrary constraints formalized in (Object Constraint Language) OCL language. In this way, the approach allows checking the syntactic correctness of a schema, both from the point of view of its definition and of its correspondence to the requirements. However, this approach requires the UML designer to be aware of the logic formalization questions of the schema.

For the semantic quality, 36.36% of the journal articles fall to this category. Semantic quality refers to how faithfully the modeled system is represented. There are two semantic goals: validity which means that all statements made in the model are correct and relevant to describe/specify the modelled system and completeness which means that the model contains all the statements which would be correct and relevant for describing or specifying the modelled system. The paper of Hoen, et. Al. [13] described SUDA, the Scenario-based UML Design Analysis approach, that can be used to rigorously analyze UML design class models that include OCL operation specifications. The study of Storrle [15] focused on the layout quality of UML diagrams. Another study [21] found out

that layout quality does impact the understanding of UML diagrams. They also found that diagram size had a significant influence to UML quality. Other researchers [22] even designed a tool called UMLx that is capable of analyzing commonly used UML diagrams.

For the pragmatic quality, 54.54% fall to this category. Pragmatic quality refers to how well the model is understood. This includes the following characteristics: maintainability, analyzability, understandability,
testability, functionality, executability, reusability,
complexity, and dependability. The study of Bernardi, et. Al. [11] focused on the dependability characteristic of the pragmatic quality. Dependability is the ability to deliver service that can justifiably be trusted. It is a term used to describe the availability performance and its influencing factors: reliability performance, maintainability performance and maintenance support performance. The study of Vargas, et. Al. [12] focused on the exploratory investigation into the impact of the use of UML on the change proneness of the implementation code. Their study focused on the maintainability side of the pragmatic quality. The researchers Merseguer and Tamburri [17], introduced DICE. DICE is a project that, following the model-driven engineering paradigm, aims to define a quality-driven framework for developing data intensive applications leveraging Big Data technologies. A key asset of DICE is the so called DICE profile, which offers the ability to design such application using UML and a set of additional stereotypes to characterize specific data intensive features. While other researchers [19] introduced UML modeling approach that is able to cope with change and validate non-functional properties of a system that is modeled using UML [20][23].

4 CONCLUSIONS

This paper reviews UML model quality papers published in journal articles found in ACM Digital. The research papers have been classified according to the following dimension: type of quality. Based on the results of the literature review, 9.10% of the journal articles fall to the syntactic type of quality, 36.36% fall to semantic type of quality, and 54.54% fall to pragmatic type of quality. The paper categorized in the first type of quality dealt with ensuring the correctness of the UML model. The papers categorized in the second type of quality mostly dealt with the design and layout of UML diagrams. While papers that fall in the third category focused on maintainability and the ability to cope with changes.

5 REFERENCES

[1] Bashir, R. S., Lee, S. P., Khan, S. U. R., Chang, V., & Farid, S. (2016). UML models consistency management: Guidelines for software quality manager. International Journal of Information Management, 36(6), 883-899.

[2] Jolak, R., Le, K. D., Sener, K. B., & Chaudron, M. R. (2018, March). OctoBubbles: A Multi-view interactive environment for concurrent visualization and synchronization of UML models and code. In 2018 IEEE 25th International Conference on Software Analysis, Evolution and Reengineering (SANER) (pp. 482-486). IEEE.

[3] Rumbaugh, J., Booch, G., & Jacobson, I. (2017). The unified modeling language reference manual. Addison Wesley.

[4] Krenn, W., Schlick, R., Tiran, S., Aichernig, B., Jobstl, E., & Brandl, H. (2015, April).

Momut:: UML model-based mutation testing for UML. In Software Testing, Verification and Validation (ICST), 2015 IEEE 8th International Conference on (pp. 1-8). IEEE.

[5] Bavota, G., Gravino, C., Oliveto, R., De Lucia, A., Tortora, G., Genero, M., & Cruz-Lemus, J. A. (2015). A fine-grained analysis of the support provided by UML class diagrams and ER diagrams during data model maintenance. Software & Systems Modeling, 14(1), 287-306.

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

[7] Acerbis, R., Bongio, A., Butti, S., & Brambilla, M. (2015, May). Model-driven development of cross-platform mobile applications with webratio and ifml. In Proceedings of the Second ACM International Conference on Mobile Software Engineering and Systems (pp. 170-171). IEEE Press.

[8] Rumbaugh, J., Booch, G., & Jacobson, I. (2017). The unified modeling language reference manual. Addison Wesley.

[9] Deryugina, O. (2016). Improving the structural quality of UML class diagrams with the genetic algorithm. In ITM Web of Conferences (Vol. 6). EDP Sciences.

[10] Krogstie, J., Lindland, O. I., & Sindre, G. (2016). Defining quality aspects for conceptual. Information System Concepts: Towards a consolidation of views, 215.

[11] Casale, G., Ardagna, D., Artac, M., Barbier, F., Di Nitto, E., Henry, A., ... & Pérez, J. F. (2015, May). DICE: quality-driven development of data-intensive cloud applications. In Modeling in Software Engineering (MiSE), 2015 IEEE/ACM 7th International Workshop on (pp. 78-83). IEEE.

[12] Zozas, I., Bibi, S., Katsaros, D., Bozanis, P., & Stamelos, I. (2017, August). Feature models for big data applications: Modeling big data applications by applying feature models. In Information, Intelligence, Systems & Applications (IISA), 2017 8th International Conference on (pp. 1-6). IEEE.

[13] Duarte, R., da Silveira, D. S., Araújo, J., & Wanderley, F. (2016, June). Towards a non-conformity detection method between conceptual and business process models. In Research Challenges in Information Science (RCIS), 2016 IEEE Tenth International Conference on (pp. 1-6). IEEE.

[14] Gogolla, M., & Doan, K. H. (2017). Quality Improvement of Conceptual UML and OCL Schemata through Model Validation and Verification. In Conceptual Modeling Perspectives (pp. 155-168). Springer, Cham.

[15] Fernández-Sáez, A. M., Genero, M., Chaudron, M. R., Caivano, D., & Ramos, I. (2015). Are Forward Designed or Reverse-Engineered UML diagrams more helpful for code maintenance?: A family of experiments. Information and Software Technology, 57, 644-663.

[16] Lethbridge, T. C., Abdelzad, V., Orabi, M. H., Orabi, A. H., & Adesina, O. (2016, October). Merging modeling and programming using Umple. In International Symposium on Leveraging Applications of Formal Methods (pp. 187-197). Springer, Cham.

[17] Gómez, A., Merseguer, J., Di Nitto, E., & Tamburri, D. A. (2016, July). Towards a UML profile for data intensive applications. In proceedings of the 2nd International Workshop on Quality-Aware DevOps (pp. 18-23). ACM.

[18] Casale, G., Ardagna, D., Artac, M., Barbier, F., Di Nitto, E., Henry, A., ... & Pérez, J. F. (2015, May). DICE: quality-driven development of data-intensive cloud applications. In Modeling in Software Engineering (MiSE), 2015 IEEE/ACM 7th International Workshop on (pp. 78-83). IEEE.

[19] Granchelli, G., Cardarelli, M., Di Francesco, P., Malavolta, I., Iovino, L., & Di Salle, A. (2017, April). Towards recovering the software architecture of microservice-based systems. In Software Architecture Workshops (ICSAW), 2017 IEEE International Conference on (pp. 46-53). IEEE.

[20] Guermazi, S., Tatibouet, J., Cuccuru, A., Dhouib, S., Gérard, S., & Seidewitz, E. (2015). Executable modeling with fUML and alf in papyrus: tooling and experiments. strategies, 11, 12.

[21] Ogata, S., & Matsuura, S. (2012, February). Training of requirements analysis modeling with UML-based prototype generation tool. In Proceedings of the 5th India Software Engineering Conference (pp. 105-108). ACM.

[22] Qi, K., & Boehm, B. W. (2018, May). UMLx: a UML diagram analytic tool for software management decisions. In Proceedings of the 40th International Conference on Software Engineering: Companion Proceeedings (pp. 278-279). ACM.

[23] Laurent, Y., Bendraou, R., & Gervais, M. P. (2013, March). Executing and debugging UML models: an fUML extension. In Proceedings of the 28th Annual ACM Symposium on Applied Computing (pp. 1095-1102). ACM.

[24] Costa, A., Cavalheiro, S., Foss, L., & Ribeiro, L. (2015, April). From UML diagrams to simulink models: a precise and verified translation. In Proceedings of the 30th Annual ACM Symposium on Applied Computing (pp. 1547-1552). ACM.