A Case of Industrial vs. Open-Source OCL: Not So Different After All

Josh G.M. Mengerink
Dep. Mathematics & Computer Science
Eindhoven University of Technology
The Netherlands
j.g.m.mengerink@tue.nl

Jeroen Noten
Dep. Mathematics & Computer Science
Eindhoven University of Technology
The Netherlands
j.f.h.noten@student.tue.nl

Ramon R.H. Schiffelers
Dep. Mathematics & Computer Science
Eindhoven University of Technology
The Netherlands
r.r.h.schiffelers@tue.nl

Mark G.J. van den Brand
Dep. Mathematics & Computer Science
Eindhoven University of Technology
The Netherlands
m.g.j.v.d.brand@tue.nl

Alexander Serebrenik
Dep. Mathematics & Computer Science
Eindhoven University of Technology
The Netherlands
a.serebrenik@tue.nl

***Abstract*—When studying model-driven engineering (MDE) in industry, generalization of studies is often hard, as most artifacts are proprietary and confidential in nature. A possible solution would be to study open-source artifacts. However, open-source artifacts are not necessarily representative for those found in the industry.**

As the first step towards investigating the viability of open-source MDE artifacts as an alternative to less accessible industrial ones, we use a large open-source dataset and several industrial meta-models to show that the complexity of OCL expressions in open-source and industry is similar.

I. Motivation and goals

Model-Driven Engineering (MDE) is being used in Industry to assist engineers in specifying systems [1], [2]. By using MDE to create domain-specific languages (DSLs), engineers can specify these systems in terms relative to their domain, rather than encoding them into general-purpose languages. However, the metamodels that underpin these DSLs are often highly complex, and at some point their expressive power is not sufficient to accurately model the domain. For instance, type systems require extra expressive power [3].

To mitigate this deficiency in metamodels, more complex mechanisms such as the Object Constraint Language (OCL) [4] have been proposed. OCL allows DSL engineers to write down complex constraints on valid models, such that the domain can be modeled more accurately. OCL has been subject to many studies in a variety of contexts such as usage [5], [6], [7], verification [8], [9], and maintenance [10]. Several of these studies have already concluded that lack of data might threaten generalizability of their conclusions [5], [10]. In particular, this lack of data holds for studies on industrial data, as most industrial applications of MDE (and thus OCL) are proprietary (and thus confidential) in nature.

We envision that open-source can be used as means to demonstrate and evaluate practical limitations of techniques proposed to analyze [11], [12] and visualize OCL [13]. For open-source it is easier to create large and publicly available datasets [5], [7] to ensure generalization and replication of results.

In order to be able to evaluate techniques on open-source artifacts and derive conclusions valid for the industry, there should be sufficient evidence that open-source artifacts can be seen as representative of industrial practice. While similar observations have been made for non-MDE software [14], it is not *a priori* clear that this also the case for OCL. Hence, a plethora of measurements should be performed to test for differences between the open-source MDE artifacts and their industrial counterparts. As a first step, in this work, we test whether complexity of open-source OCL expressions differs from complexity of the industrial ones. We have chosen to start with complexity, as it encompasses various aspects of artifacts. As such, it should serve as a good indication of similarity, or difference between open-source and industry.

In our previous work [7] we have constructed a publicly available dataset of over 9000 OCL expressions. We compare this dataset with the data obtained from the industry, and ask the following research question:

> Do the complexities of open-source and industrial OCL code differ?

II. Data Description and Analysis

We analyze a dataset of OCL expressions[1] previously mined from open source GitHub projects [7], and a dataset of OCL expressions from industrial projects by ALTRAN. The GitHub dataset includes `.ocl` and `.ecore` files (`.ecore` files are included as they may have embedded OCL expressions). It contains over 9000 OCL expressions obtained from those files, i.e., more than ten times more than datasets used in previous studies [5] and includes the dataset of Cabot[2].

[1]https://github.com/tue-mdse/ocl-dataset
[2]https://github.com/jcabot/ocl-repository

The ALTRAN dataset is derived from seven metamodels obtained from ALTRAN, a large company offering third-party MDE services. Using EMMA, our EMF (Meta)Model Analysis tool [15], we extracted 73 OCL expressions.

To compare the datasets we focus on complexity. Complexity is one of the most studied aspects of software quality both in MDE- and traditional software [16], [17], [18]. For OCL expressions complexity has been operationalized as "the number of distinct properties" used by an expression [5]. For instance the expression "**context** Person **inv:** self.age $>=$ 0" has a complexity of one, as it only references the age property of Person. On the other hand, the expression "**context** Auto **inv:** self.registration $>=$ self.constructionYear" has a complexity of two as it references both the registration, and constructionYear.

In order to determine whether the complexities of open-source and industrial OCL code differ, we apply a Mann-Whitney-Wilcoxon test [19]. We opt for this test since it is non-parametric [20], i.e., does not make assumptions about the shape of the underlying distributions, and is robust in presence of populations of unequal sizes [19]. Moreover, it is commonly used in software engineering research [21]. As null-hypothesis (H_0) we take therefore: "The distributions of complexity of the samples of industrial and open-source OCL expressions represent two populations with the same median values" , leaving the alternative hypothesis (H_a) to be: "The distributions of complexity of the samples of industrial and open-source OCL expressions represent two populations with different median values". To reject the null hypothesis we use the traditional threshold of 0.05.

III. Results and Discussion

We start by inspecting Figure 1. It shows a violin plot [22] of the computed complexities. The median, Q3, and maximum complexity of open-source OCL expressions from GitHub are higher (2, 3, 36, respectively) than those of the industrial expressions from the ALTRAN dataset (1, 2, 5, respectively).

Statistical comparison of the distributions, however, results in the p-value of the Mann-Whitney-Wilcoxon test being 0.05591, which slightly exceeds the traditional threshold of 0.05. Hence, as far as expression complexity is concerned, the differences observed above are not enough to claim that the complexity distributions are statistically different. There is no reason to assume that the industrial OCL expressions differ from open-source OCL expressions.

We can conclude, thus, that future results obtained for the open-source OCL expressions are likely to be valid for industrial OCL expressions as well.

Validity of the previous conclusion might have been threatened by the limited size of the ALTRAN dataset that may not be representative of industrial practice in general. Due to the proprietary nature of industrial models, there is little we can do about this. However, as the open-source dataset is publicly available,[3] we encourage the reader to replicate our study on their proprietary datasets.

[3]https://github.com/tue-mdse/ocl-dataset

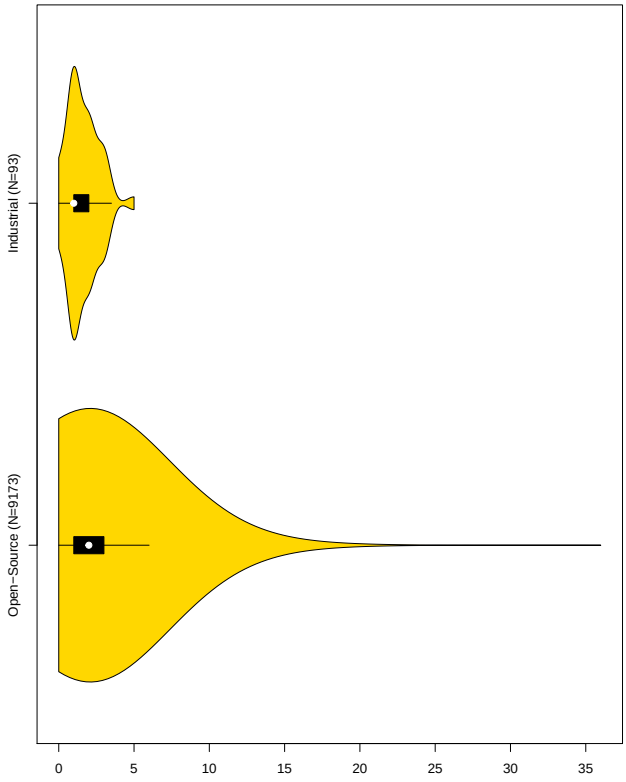

Fig. 1: Open-source OCL expressions appear to be slightly more complex than the industrial ones.

A concern often raised with data mined from GitHub is that some data may merely be examples rather than "real" artifacts (cf. [23]). We inherit this threat from the previous work of Noten *et al.* [7]. Of the 16502 Ecore files in this dataset, 3280 contained the word "example" in their path; for OCL files, 150 of 890. Circa 20% of the dataset files are, hence, examples.

IV. Conclusions

In this work we suggest applying MDE techniques to (widely available) open-source data, rather than (scarce) industrial data. In particular, we have focused on the Object Constraint Language (OCL).

As a first step to verifying whether open-source data can be used as a proxy for industrial data, we have compared the distributions of complexity among OCL expressions. We have found that *complexity of OCL expressions does not differ between our industrial and open-source datasets*.

Complexity is only the first step in evaluating techniques on open-source data. Thus, as future work, we envision performing similar studies for a large variety of properties. For instance, distribution of used constructs and multiplicities, or their evolution. Additionally, we encourage the reader to perform new experiments on the dataset of Noten *et al.* [7].

REFERENCES

[1] Herrmannsdörfer, M., Benz, S., Juergens, E.: Automatability of coupled evolution of metamodels and models in practice. In: MoDELS. Springer (2008) 645–659

[2] Schiffelers, R.R.H., Alberts, W., Voeten, J.P.M.: Model-based specification, analysis and synthesis of servo controllers for lithoscanners. In: 6th International Workshop on Multi-Paradigm Modeling, ACM (2012) 55–60

[3] van den Brand, M.G.J., van der Meer, A.P., Serebrenik, A., Hofkamp, A.T.: Formally specified type checkers for domain specific languages: experience report. In: LDTA, ACM (2010) 12

[4] Warmer, J., Kleppe, A.: The Object Constraint Language: Getting Your Models Ready for MDA. 2 edn. Addison-Wesley (2003)

[5] Cadavid, J.J., Combemale, B., Baudry, B.: An analysis of metamodeling practices for MOF and OCL. Computer Languages, Systems & Structures **41** (2015) 42–65

[6] Kolovos, D.S., Matragkas, N.D., Korkontzelos, I., Ananiadou, S., Paige, R.F.: Assessing the use of eclipse mde technologies in open-source software projects. In: OSS4MDE@ MoDELS. (2015) 20–29

[7] Noten, J., Mengerink, J.G.M., Serebrenik, A.: A data set of OCL expressions on GitHub. In: MSR. (2017)

[8] González, C.A., Büttner, F., Clarisó, R., Cabot, J.: EMFtoCSP: A tool for the lightweight verification of EMF models. In: Formal Methods in Software Engineering: Rigorous and Agile Approaches, IEEE (2012) 44–50

[9] Richters, M., Gogolla, M.: On formalizing the UML object constraint language OCL. In: Conceptual Modeling. (1998) 449–464

[10] Khelladi, D.E., Hebig, R., Bendraou, R., Robin, J., Gervais, M.P.: Metamodel and constraints co-evolution: A semi automatic maintenance of OCL constraints. In: ICSR, Springer (2016) 333–349

[11] Anastasakis, K., Bordbar, B., Georg, G., Ray, I.: On challenges of model transformation from UML to Alloy. Software & Systems Modeling **9**(1) (2008) 69

[12] Kuhlmann, M., Hamann, L., Gogolla, M.: Extensive validation of OCL models by integrating SAT solving into USE. In: TOOLS, Springer (2011) 290–306

[13] Bottoni, P., Koch, M., Parisi-Presicce, F., Taentzer, G.: A visualization of ocl using collaborations. In Gogolla, M., Kobryn, C., eds.: UML. Springer (2001) 257–271

[14] Hunsen, C., Zhang, B., Siegmund, J., Kästner, C., Leßenich, O., Becker, M., Apel, S.: Preprocessor-based variability in open-source and industrial software systems: An empirical study. Empirical Software Engineering **21**(2) (2016) 449–482

[15] Mengerink, J.G.M., Serebrenik, A., Schiffelers, R.R.H., van den Brand, M.G.J.: Automated analyses of model-driven artifacts: Obtaining insights into real-life application of MDE. In: IWSM Mensura. (2017)

[16] Gerpheide, C.M., Schiffelers, R.R.H., Serebrenik, A.: Assessing and improving quality of QVTo model transformations. Software Quality Journal **24**(3) (2016) 797–834

[17] Landman, D., Serebrenik, A., Bouwers, E., Vinju, J.J.: Empirical analysis of the relationship between CC and SLOC in a large corpus of java methods and C functions. Journal of Software: Evolution and Process **28**(7) (2016) 589–618

[18] Olszewska, M., Dajsuren, Y., Altinger, H., Serebrenik, A., Waldén, M.A., van den Brand, M.G.J.: Tailoring complexity metrics for simulink models. In: ECSA Workshops, ACM (2016) 5

[19] Mann, H.B., Whitney, D.R.: On a test of whether one of two random variables is stochastically larger than the other. Ann. Math. Statist. **18**(1) (03 1947) 50–60

[20] Sheskin, D.J.: Handbook of parametric and nonparametric statistical procedures. CRC Press (2003)

[21] Dybå, T., Kampenes, V.B., Sjøberg, D.I.: A systematic review of statistical power in software engineering experiments. IST **48**(8) (2006) 745 – 755

[22] Hintze, J.L., Nelson, R.D.: Violin plots: A box plot-density trace synergism. The American Statistician **52**(2) (1998) 181–184

[23] Kalliamvakou, E., Gousios, G., Blincoe, K., Singer, L., Germán, D.M., Damian, D.: The promises and perils of mining GitHub. In: MSR. (2014) 92–101