

Evaluation of the Effectiveness of Domain Modeling Methods in Terms of Model Complexity

Iryna Martinkus¹ [0000-0003-4629-6583], Heinrich C. Mayr² [0000-0001-5770-8091],
Kostiantyn Nagornyj¹ [0000-0001-5948-3682], Mykola Tkachuk¹ [0000-0003-0852-1081]

¹ V.N. Karazin Kharkiv National University, Majdan Svobody 4, Kharkiv, Ukraine 61077

² Alpen-Adria-Universität Klagenfurt, Universitätsstraße 65-67, A-9020.Austria
imartinkus@gmail.com, heinrich.mayr@aau.at,
k.nagornyj@gmail.com, tka.mobile@gmail.com

Abstract. Domain Modeling Methods (DMMs) are used to improve quality in the development of complex software systems such as product families. This paper examines how the effectiveness of such methods in combination with appropriate case tools (CTs) can be evaluated with regard to (1) the complexity of a model to be created and (2) the reusability of the developed source code. We formalize an effectiveness coefficient and its computation using structured data resources and quantitative metrics. The proposed approach is illustrated by analyzing and comparing the DMMs JODA (Joint integrated Object oriented Domain Analysis) and ODM (Organizational Domain Modeling) as well as the CTs Actifsource and Eclipse Modeling Framework.

Key words: domain modeling method, software, code reuse, effectiveness, metric

1 Introduction: Problem Actuality and Research Aim

The main goals of modern software engineering are to provide a quality of the developed product with regard to the robust realization of all requirements, cost effectiveness as well as the shortest possible time to market. One approach to achieving these goals is Model-Driven Software Development (MDSD) [1], which aims to generate executable source code from a domain model (DM) to be created using an appropriate DMM. This raises the question for a concrete software project: which of the available DMMS is most effective for a given project? The assessment of an effectiveness of DMMs is widely discussed in the literature, where DM complexity and reusability of generated source code are key criteria. In [2], results of an empirical study on code reuse (CR) trends in open source software are presented, and a collection of reuse metrics to define the most important CR factors is proposed. [3] focuses on constructing a relationship between CR extent (CRE) and values of software complexity metrics in object-oriented programming. A more detailed analysis of these issues can be found in [4].

The aspect of model complexity and its estimation, for example, is dealt with in [5] analyzing all structural model components like objects, relations, types and number of

operations (methods). [6] presents simple metrics that allow, for a given DM, to estimate the complexity of its interface, elements, and attributes, and suggests a combination of these three values to obtain a total complexity index. [7] proposes an advanced approach to assessing DM complexity in terms of both, functionality and dynamic aspects. But only a few studies address (e.g. [8]) the relationship between DM complexity and DMMs effectiveness, however, is rarely analyzed. We present an approach to answering this question, which is essentially based on a special metric: “effectiveness coefficient ratio (ECR)” that can be used to assess the effectiveness of alternative DMMS and CTs in terms of DM complexity and code reusability.

2 An Approach to Evaluation

Within the last 10-15 years, a variety of DMMs have been developed, and one of their possible classification is presented in [4]. We examined two of these methods in more details: JODA and ODM. JODA (Joint Object oriented Domain Analysis) uses the object-oriented approach to cover the domain analysis phase. ODM (Organizational Domain Modeling) systematically supports the mapping of domain-specific artifacts into project assets that can be reused in future software development activities.

Obviously, any DMM without a supporting CT is only of limited use in software development practice. There are many such CTs with very different capabilities, and their comparative analyse is also shown in [4]. For the purpose of our research, such CTs as Actifsource and Eclipse Modeling Framework were chosen.

One of the aspects to be considered here is the question of whether one can select a DMM, and a CT for a given DM in such a way that one obtains the most effective solution possible. For this purpose, we introduce a collection of heuristics, denotations, and definitions which we have taken from the literature or gained from our own experience of DDD-oriented software development:

- We denote a certain domain modeling method by μ_i , a certain case tool by γ_i .
- We call a pair $\tau_{ij}=(\mu_i \gamma_i)$ a *Domain Modeling Technique*, if γ_i supports μ_i , and we assume that a τ_{ij} ensures consistency of any Model $M_{\tau_{ij}}$ created using it.
- Furthermore, we assume that we can associate to each (consistent) Model $M_{\tau_{ij}}$ a code framework $\Phi(M_{\tau_{ij}})$ using an appropriate generator.
- The *Complexity Level* $DMC(M_{\tau_{ij}})$ of a model and the *Reusability Extent* $CRE(\Phi(M_{\tau_{ij}}))$ of a code framework is expressed by a positive value. The algorithm for calculating these values is presented in Section 3.2.

Based here-on, we define the *Effectiveness Coefficient* $E(M_{\tau_{ij}})$ of using a technique τ_{ij} for creating a model $M_{\tau_{ij}}$ by the quotient

$$E(M_{\tau_{ij}}) = CRE(\Phi(M_{\tau_{ij}})) / DMC(M_{\tau_{ij}}), \quad (1)$$

and we define an *Effectiveness Coefficient Ratio* **ECR** of two techniques τ_a and τ_b for a given domain model M with $E(M_{\tau_a}) > E(M_{\tau_b})$ as the quotient:

$$ECR_{M,\tau_a,\tau_b} = \frac{\Delta(E(M\tau_a), E(M\tau_b))}{E(Mt_b)} \quad (2)$$

The computation of the reusability extent **CRE(Φ)** utilizes the well-known software structural complexity metrics [3]: WMC (Weighted Methods per Class), RFC (Responses for a Class), DIT (Depth of Inheritance Tree), NOC (Number of Children), CBO (Coupling between Object). In [4] the follows formula is constructed

$$CRE = 0,1198 * WMC + 0,0398 * RFC + 0,2801 * DIT + 0,3603 * NOC + 0,2000 * CBO \quad (3)$$

The calculation of the complexity level of a given model M supposes the calculation of complexity values for each class included in this M, the complexity values of all relationships between classes in M, and finally the overall complexity level value **DMC(M)**. The appropriate expression for this purpose is the follows [7]:

$$DMC = K_C * (\#Class) * CC + K_{RC} * RC; K_C = 0.7; K_{RC} = 0.3 \quad (4)$$

where (#Class) is the number of DM classes, CC (ClassComplexity) is a value that determines quantitatively the overall structural and functional class's complexity, RC as summarized value of relationships complexity, and the weight coefficients 0.7 and 0.3 have been defined with the appropriate expert-oriented procedure with usage of Analytic Hierarchy Process [9].

3 Case Study: Experimental Results and Discussion

The framework for evaluating the proposed approach has been elaborated in [7]. In this context two use case DMs were developed for the Universe of Discourse «Students Personal Data Processing in Education Management System» using the methods ODM and JODA, and the CASE-tools EMF and Actifsource respectively. In this publication we present the new experimental results to prove the proposed approach to assessment of effectiveness coefficient ratio **ECR**.

We have defined 8 test cases with different numbers of domain objects (see Table 1), and constructed the corresponding DMs using the both modeling technologies.

Table 1 Test variants

№	Domain Objects
Test 1	11
Test 2	10
Test 3	19
Test 4	15
Test 5	12
Test 6	19
Test 7	27
Test 8	34

This led to the results presented in Table 2. They show that the applying the ODM/EMF technology to all test cases provided more effective than the JODA/Actifsource technology.

Table 2. Experimental results for the test cases

	$M_{\tau ODM,EMF}$			$M_{\tau JODA,Actifsource}$			ΔE	ECR
	DMC	CRE	E	DMC	CRE	E		
Test 1	82,52	20,52	0,25	107,53	9,08	0,08	0,17	1,89
Test 2	64,04	15,34	0,24	87,09	7,22	0,08	0,16	1,81
Test 3	30,91	11,78	0,38	30,46	4,87	0,16	0,22	1,38
Test 4	56,51	12,26	0,22	91,05	7,84	0,09	0,13	1,51
Test 5	120,79	16,83	0,14	162,50	7,64	0,05	0,09	1,91
Test 6	627,08	47,60	0,08	727,00	31,14	0,04	0,04	0,70
Test 7	1071,81	83,47	0,08	1191,96	54,54	0,05	0,03	0,66
Test 8	1889,73	125,23	0,07	2058,40	78,94	0,04	0,03	0,52

To discuss this result briefly, we point out once again that the reusability extent CRE of the source code depends more on the CASE-tool used to support a given DMM, while the complexity level DMC depends more on an expertise of the modeler. To analyze the latter impact factor we considered the relation between the DMC parameter and the efficiency coefficient E (see Fig. 1) which proves that the effectiveness coefficient E decreases substantially with an increasing DM complexity.

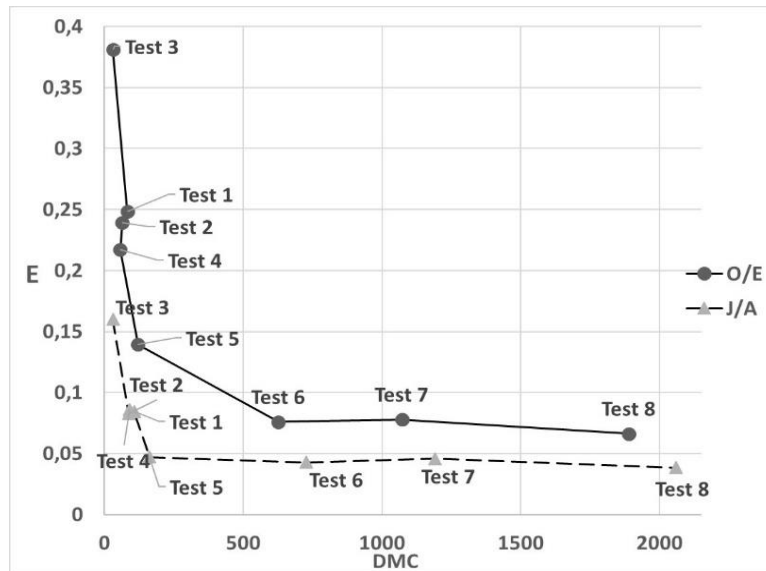


Fig. 1. The impact of complexity on the effectiveness coefficient E

This fact indicates a sufficient validity of our proposed approach to determine the effectiveness of different domain modeling techniques when applied to given domain model. Moreover, the proposed evaluation approach ensures the selection of a domain modeling technique with a favorable values of *ECR* (in 0.5-1.9 times in our tests).

Finally, we investigate the ratio of how the *ECR* is dependent on the number of domain objects (#DO) in the given DMs. Table 3 and it's visualization in Fig. 2 show that this ratio decreases, if the number of domain objects increases.

Table 3. The influence ECR/#DO ratio

	ECR	#DO	ECR/#DO
Test 1	1,89	11	0,17
Test 2	1,81	10	0,18
Test 3	1,38	19	0,07
Test 4	1,51	15	0,10
Test 5	1,91	12	0,16
Test 6	0,70	19	0,04
Test 7	0,66	27	0,02
Test 8	0,52	34	0,02

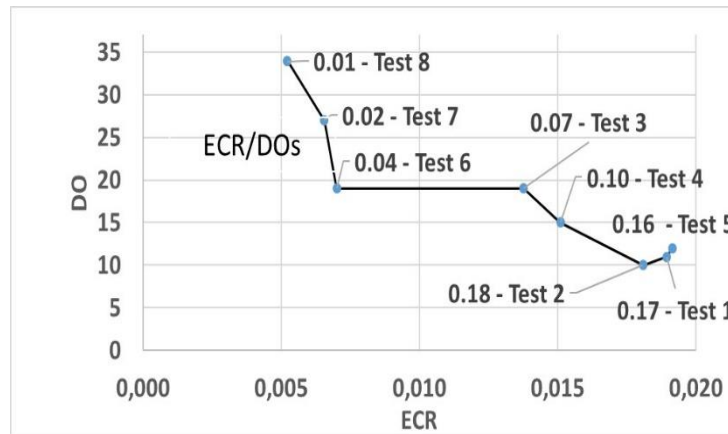


Fig. 2. Visualization of Table 3

Finally, we note that the test data computation described above were performed using a special tool [7] developed with the license-free Java and MySQL software.

4 Conclusions and Future Work

We have identified as a key variability driver for a given MDS D-project the choice of the most effective combination of a domain modeling method and a supporting CASE tool. In order to provide this selection, we have developed and presented suitable metrics (Effectiveness Coefficient *E* and Effectiveness Coefficient Ratio *ECR*) that also

consider the aspect of reusability of the generated source code. For two alternative domain modeling techniques: ODM/ JODA and EMF/Actifsource, we performed a series of tests that proved the proposed approach.

What interests us in future is how we can generalize this approach so that it provides reliable results for all models created with a Domain Specific Modeling Languages (DSMLs) [10]. Therefore, the metrics would then have to reflect the properties of the metamodels underlying the DSMLs considered, instead of the concrete domain model properties.

References

1. Tune, N.; Millet, S.: *Patterns, Principles And Practices Of Domain-driven Design*. 1st ed., John Wiley & Sons (2015)
2. Taibi, F.: Empirical Analysis of the Reusability of Object-Oriented Program Code in Open-Source Software. In: *Int. Journal of Computer, Electrical, Automation, Control and Information Engineering*, Vol.8, No.1 (2014)
3. Nandakumar, A.N.: Constructing Relationship between Software Metrics and Code Reusability in Object Oriented Design. In: *Int. Journal of Advanced Computer Science and Applications*, Vol. 7, No. 2 (2016)
4. Tkachuk, M.; Martinkus, I.; Gamzayev, R. et al.: An Integrated Approach to Evaluation of Domain Modeling Methods and Tools for Improvement of Code Reusability in Software Development. In (Mayr, H.C.; Pinzger, M. eds.): *INFORMATIK 2016, Lecture Notes in Informatics*, Vol. P-259: Köllen Verlag, Bonn, pp. 143--156 (2016)
5. Preschern, C.; Kajtazovic, N.; Kreiner C.: Evaluation of Domain Modeling Decisions for two identical Domain Specific Languages. In: *Lecture Notes on Software Engineering*, Vol. 2, No. 1, February, pp. 37--41 (2014)
6. Leitner, A.; Weiss, A.; Kreiner C.: Analyzing the Complexity of Domain Model Representations In: *IEEE 19th International Conference and Workshops on Engineering of Computer-Based Systems*, Novi Sad, Serbia (2012)
7. Tkachuk M.; Gamzaev R.; Martinkus I. et al.: Towards Effectiveness Assessment of Domain Modelling Methods and Tools in Software Product Lines Development. In: *Enterprise Modelling and Information Systems Architectures, Int. J. of Conceptual Modelling*, Vol. 13, pp. 190--206 (2018)
8. Nisa, H.; Khan, M.: Impact of Domain Modeling Techniques on the Quality of Domain Model: An Experiment. In: *Int. J. of Advanced Computer Science and Applications*, Vol. 7, No. 11, pp. 454--462 (2016)
9. Saaty T. L.: *Fundamentals of the Analytic Hierarchy Process*. RWS Publishing (2000)
10. Karagiannis, D.; Mayr, H.C.; Mylopoulos, J.: *Domain-Specific Conceptual Modeling: Concepts, Methods and Tools*. Springer, Berlin (2016)