MoCoP: Towards a Model Clone Portal

Önder Babur
Dept. of Mathematics & Computer Science
Eindhoven University of Technology
Eindhoven, The Netherlands
o.babur@tue.nl

Matthew Stephan
Dept. of Computer Science & Software Engineering
Miami University
Oxford, Ohio, USA
stephamd@miamioh.edu

***Abstract*—Widespread and mature practice of model-driven engineering is leading to a growing number of modeling artifacts and challenges in their management. Model clone detection (MCD) is an important approach for managing and maintaining modeling artifacts. While its counterpart in traditional source code development, code clone detection, is enjoying popularity and more than two decades of development, MCD is still in its infancy in terms of research and tooling. We aim to develop a portal for model clone detection, MoCoP, as a central hub to mitigate adoption barriers and foster MCD research. In this short paper, we present our vision for MoCoP and its features and goals. We discuss MoCoP's key components that we plan on realizing in the short term including public tooling, curated data sets, and a body of MCD knowledge. Our longer term goals include a dedicated service-oriented infrastructure, contests, and forums. We believe MoCoP will strengthen MCD research, tooling, and the community, which in turn will lead to better quality, maintenance, and scalability for model-driven engineering practices.**

***Index Terms*—model-driven engineering, model clone detection, model analytics, software maintenance, model management, model repositories.**

I. Introduction

The increasing volume and complexity of software systems has led academia and industry to explore methodologies to better facilitate the engineering and maintenance of those systems. Model-driven engineering (MDE) is one such methodology. However, as MDE practices mature, the volume and complexity of its corresponding artifacts, notably models, transformations, and modeling languages is increasing in kind. This manifests itself in industrial practice in individual companies [1], [2], as well as in the open source domain, and academic repositories [1], [3], [4]. Taking also model evolution into account, large-scale MDE practitioners have to deal increasingly with legacy models and ecosystems [5]. This necessitates scalable techniques for model management; in particular, organizing, searching, reusing, maintaining, analyzing, synthesizing, and visualizing these large and complex set of artifacts and ecosystems.

Management of MDE artifacts has manifested itself as a crucial focus of both emerging model repositories and model management research. The latter involves a wide range of techniques. Model management is a high-level concept in which entire models and their relationships can be manipulated using operators to achieve useful outcomes [6], [7] or made consistent with respect to other models [8]. Issues within model management are addressed by researchers through increasingly popular model portals/repositories, which are an important step and challenge in furthering model-driven solutions [9]. Model repositories allow convenient web-based interfaces to store, version control, retrieve, and analyze MDE artifacts.

Model clone detection (MCD) is one form of model management. Models, or their fragments, which are similar according to some similarity measure, are identified by model clone detectors to infer valuable insights into understanding underlying ecosystems, their quality, maintainability, and more. The analogous source code clone detection (CCD) has been experiencing popularity for over two decades [10]. CCD, an established form of source code analysis and manipulation, is very important in software development. In contrast to CCD, there has been only limited volume of research and a lack of much available, usable, and mature tooling for MCD. There has even been a drop in MCD research in recent years despite the benefits it has provided to industry [11].

There are several challenges hampering the advancement and adoption of MCD techniques. These challenges include,

- A lack of available, usable, mature, open source tooling for MCD
- Approaches/tools that are specific to certain modeling languages, and difficult to apply to different types of models
- Having no curated data sets, standard corpora or benchmarks, and only a limited set of comparative studies
- There being no repository or portal allowing others to experiment/use/evaluate model clone detectors, nor any *body of knowledge* for MCD research
- End user *usability* and researcher *reproducibility*

In this paper, we propose tackling these challenges and promoting MCD research and practice through creation of a model clone portal (MoCoP) for disseminating MCD tools, data sets, knowledge, and more. We begin the paper with Section II by providing background on MCD concepts, literature, and tools. Section III presents related model repositories and initiatives. We describe the important features of MoCoP and provide illustrations of it in Section IV. We conclude the paper and discuss next steps in Section V.

II. Background - Model Clone Detection

MCD is a form of model comparison that involves identifying sets of similar model elements [12]. Model clone detectors measure similarity in a variety of ways including graphical

analysis [13] or inference based on models' underlying textual representations [14]. Analogously to traditional CCD, there are multiple types of model clones to detect [14], [15]: 1) type 1, identical model clones; 2) type 2, renamed model clones; 3) type 3, near-miss model clones; and 4) type 4, semantically equivalent clones. MCD is available for multiple different modeling languages, with Simulink being the most prevalent [13], [14], [16]. UML models are also a growing target for MCD [15], [17]. More recently, researchers are working on approaches for MCD in EMF metamodels [18], and model transformation languages [19]. The information provided by model clone analysis includes pattern clustering [11], [20], anti pattern detection [21], security and quality analysis [22], [23], and more. Evaluation techniques for model clone detectors are relatively sparse [24], which is one of the benefits of creating a portal as proposed in this paper.

A. Volume of MCD Research and Available Tooling

Compared to CCD, MCD is not getting much attention in the literature. For instance, the venue *International Workshop on Software Clones* with its 13 iterations contains an overwhelming majority of CCD papers, while only a handful of MCD papers. Additionally, MCD research and tooling seems to be in decline in recent years; and there is only limited *focused* knowledge about the core of the concepts, elements, mechanisms and practices of MDE (except, for instance, the one by Stephan et al. [12]). In pursuit of building an MCD body of knowledge, similar in style, but more focused than the in-progress Model-Based Software Engineering Body of Knowledge [25], as a long term goal, we have been performing a systematic literature study. Considering the scope of this paper, we provide some early metrics that quantitatively support our concerns about MCD research. Covering other repositories for other technical spaces; for example, database schemas and ontologies; and complementary techniques, such as snowballing, is future work.

For the data in this paper, we follow the basic protocol of (1) searching in Google Scholar using the string {*"clone detection" AND (model OR diagram OR design OR pattern)*} over the years 1999-2018, (2) manually inspecting each search result with respect to our inclusion criteria and (3) reporting the numbers. Our inclusion criteria include papers that (a) can be scientific articles or theses, (b) explicitly address models or comparable artifacts such as diagrams as first class data entities, and (c) explicitly contribute to model clone detection literature. These criteria led to the exclusion of, for instance, code clone detection techniques using an intermediate model extracted from the code, and model comparison techniques with no explicit/direct application presented in MCD context.

Figure 1 presents the numbers from our preliminary investigation. The Y-axis represents the number of search results and papers: the grey bars represents the total number of search results we retrieved from Google Scholar, and the black bar represents the total number of articles we identified as MCD literature via manual inspection. We see the number of search results undergoes an upward trend across the years. Since most of the results involved code clone detection, and therefore were excluded by us in our survey, we suspect code clone detection literature is increasing in volume. MCD literature emerges in 2006, has relatively small volume, and experiences decline after its peak in 2014. We also marked the introduction of prominent tools along the years. We identify some adoption obstacles with the tools: CloneDetective/ConQAT (2008) is discontinued and replaced with a commercial tool for non-academic use; ModelCD (2009) is not available at all; Simone (2012) is available upon request and through a FreeBSD license; MQlone (2013) is available closed-source and with very limited functionality in the MACH toolset; and SAMOS (2018) is not published nor mature yet.

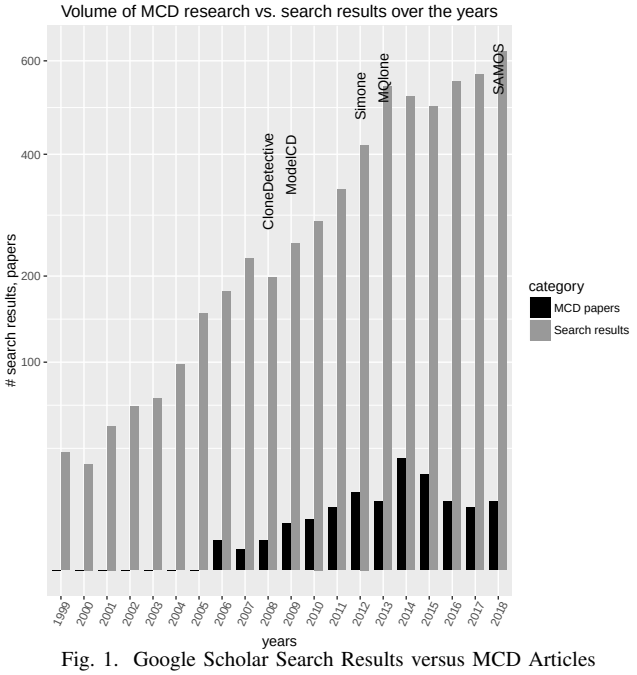

Fig. 1. Google Scholar Search Results versus MCD Articles

III. Related Work - Software Model Portals and Other Initiatives

In this section we describe other portals and related initiatives from which we draw inspiration.

A. Software Model Portals

- The open models initiative aims to provide a platform to develop conceptual and other types of models collaboratively for open public development, use, and distribution [26]. It acts as a central hub for the modeling community with an infrastructure supporting model development, teaching, and research. The lack of open source software development practices in modeling is observed by the authors. They plan to mitigate "reinventing the wheel" by providing better tried and tested models, domain expert knowledge, easy integration and reuse. The open model approach also serves as a test-bed for investigating the effects of conceptual modeling and open models on open source software development.

- MDEForge is an extensible modeling framework to support the discovery and reuse of existing modeling artifacts such as models, transformations, and domain specific languages [4]. The lack of these in current MDE practice has been identified by the authors as an issue hampering wide adoption of MDE, leading to the development of these artifacts from scratch. They present an extensible service-based repository for storing, querying, managing and reusing modeling artifacts in a convenient manner via a REST API. It supports the modeling community with the modular and collaborative nature of the repository, where new artifacts and tools, in the form of services, can be integrated easily into the repository.
- Repository for Model-Driven Development (ReMoDD) is a repository for improving MDE research, productivity, and learning [27]. It incorporates MDE case studies using graphical and specification models. It includes other related artifacts such as reusable transformations, source code of models' implementations, example models reflecting good and bad practices, exercises and other pedagogical materials for teaching modeling, and benchmark models for the testing and evaluation of MDE techniques. ReMoDD further provides a web-based interface for searching and browsing the repository, as well as a forum for the community interaction.
- models-db provides a UML repository for UML models, images, and design metrics with search functionalities via the web interface. It is intended to aid researchers in performing empirical experiments on the models. It currently hosts the Lindholmen dataset [3].

B. Other Initiatives

- Apromore process analytics portal: Apromore addresses the increasing number of process models being created by organizations, and the challenges in dealing with consulting, updating, and reusing these collections of models over long periods of time by various stakeholders [28]. It provides facilities to analyze, maintain, and exploit process models using advanced techniques such as MCD. It has an open source service-oriented architecture for usability and extendability by the modeling community.
- CCD benchmarks: Benchmarks are considered as means to advance the state of the art in software engineering [29]. The idea of using curated data sets and benchmarks in CCD research for comparing and evaluating the tools has been explored to some extent by various researchers as well [30]. Several benchmarks exist in the literature, ranging from one by Bellon et al. [31] to BigCloneBench [32] from Roy et al. The latter contains more than eight million manually validated clones within over 25,000 open-source Java systems. The former triggered various empirical studies, such as the one by Charpentier et al. on clone classification and manual labelling [33].
- Data Science and Machine Learning portals: Another related platform is *kaggle*[1], a very popular crowd-sourced platform to attract, train, and challenge data scientists. It provides data sets, contests, and an overall medium for researchers and practitioners to interact, cooperate, and compete in solving problems in data science, machine learning, and predictive analytics. It has tremendous success in the data science community, including industry, with over one million members, numerous challenges and solutions, and high community interaction.

IV. MoCoP: The Model Clone Portal

We propose the MoCoP to address the issues we identified in MCD and to foster MCD research and practice. We intend for it to act as a central hub for MCD research with practical information, pointers to literature, tooling, and a platform for community interaction. Although inspired by existing model repositories and initiatives, our vision is very much focused on MCD, yet going beyond the scope of individual repositories. Hence we advocate a separate portal. We plan to have the following features in the short and medium term,

1) Different tools along with configurations and documentation, presented in a convenient way. For example, dockers for easy deployment and use.
2) Facilitation of continuous growth of knowledge and tooling through implementation and revival of relevant techniques, ranging from graph isomorphism approaches and other unavailable tools from the literature.
3) Exploration of how and to what extent the state-of-the-art code clone detection tools can be adapted and enhanced for operating on models.
4) An investigation feature for existing tools for different modeling languages, concrete syntaxes and file formats, e.g. via model transformations, conversions, or bridges.
5) A repository of curated datasets, standard corpora, benchmarks for different types of models and tools.

We have made some preliminary progress towards MoCoP. Based on these features, we provide our sketch of the envisioned portal in Figure 2. We are constructing an initial body of knowledge and temporarily publishing it through our model management and analytics website[2].

Longer term features include challenges, along the lines of Kaggle competitions, transformation tool contests, and model matching challenges. Another feature is a forum for discussing various MCD topics, sharing publications, models and tools, which we believe will be very beneficial. While MoCoP is more than just a model repository, we acknowledge the specific challenges for model repositories in general as identified by Basciani et al. [34]. Thus we also wish to tackle some of them in the long run including a service-based infrastructure, in particular model clone detection as a service; advanced querying and searching for the artifacts; supporting heterogeneity and scalability; and licensing and artifact management supporting open science and reproducibility. We are in contact with the Data Science Center at Eindhoven University of Technology, and will continue to investigate infrastructure support and inter-disciplinary collaboration opportunities by the center.

[1]https://www.kaggle.com/

[2]https://modelanalytics.wordpress.com/mocop/

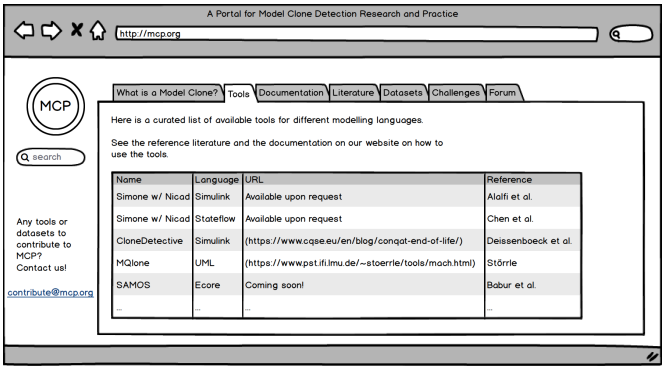

Fig. 2. Mock-up illustration of MoCoP, showing available tools.

V. Discussion and Conclusion

In this paper, we address challenges in MCD and propose a portal to help tackle them, foster MCD research and practice, and support the MCD community. The immediate next step is to realize the portal with the basic components, that is, the data, tooling, documentation and body of knowledge. As a central hub with additional input from other MCD researchers, we wish to advance the state-of-the-art in MCD through MoCoP. We are in the process of forming a body of knowledge for MCD, and will enhance it with cross-domain insights, for instance, code clone detection, approaches for other modeling languages, and technical spaces such as systems and knowledge engineering, data mining, and machine learning domains. Researchers will also be able to benefit from MoCoP in advanced studies into other important aspects of MCD, such as annotation, validation, ranking and actionability, usability, and applicability of existing techniques in industrial contexts. We are optimistic about the opportunities such a portal will afford for advancing MCD research, and look forward to receiving and incorporating feedback from the workshop and the modeling community in general.

References

[1] Ö. Babur, L. Cleophas, M. van den Brand, B. Tekinerdogan, and M. Aksit, "Models, more models, and then a lot more," in *Software Technologies: Applications and Foundations*, 2018, pp. 129–135.

[2] R. Schiffelers, Y. Luo, J. Mengerink, and M. van den Brand, "Towards automated analysis of model-driven artifacts in industry," in *MODELSWARD*, 2018, pp. 743–751.

[3] R. Hebig, T. H. Quang, M. R. Chaudron, G. Robles, and M. A. Fernandez, "The quest for open source projects that use UML: mining GitHub," in *MODELS*. ACM, 2016, pp. 173–183.

[4] F. Basciani, J. Di Rocco, D. Di Ruscio, A. Di Salle, L. Iovino, and A. Pierantonio, "MDEForge: an extensible web-based modeling platform." in *CloudMDE@ MoDELS*, 2014, pp. 66–75.

[5] M. van den Brand, "Model Driven Software Engineering creates tomorrows legacy," 2018, international Workshop on The Globalization of Modeling Languages. [Online]. Available: http://gemoc.org/pub/20181015-GEMOC18/keynote-abstract.pdf

[6] P. A. Bernstein, "Applying model management to classical meta data problems." in *CIDR*, vol. 2003, 2003, pp. 209–220.

[7] S. Kokaly, R. Salay, M. Sabetzadeh, M. Chechik, and T. Maibaum, "Model management for regulatory compliance: a position paper," in *Modeling in Software Engineering*. IEEE, 2016, pp. 74–80.

[8] W. Silva Torres, M. van den Brand, and A. Serebrenik, "Model management tools for models of different domains: a systematic literature review," in *International Systems Conference*. IEEE, 2019.

[9] R. France and B. Rumpe, "Model-driven development of complex software: A research roadmap," in *Future of Software Engineering*. IEEE, 2007, pp. 37–54.

[10] C. K. Roy and J. R. Cordy, "A survey on software clone detection research," *Queens School of Computing TR*, vol. 541, no. 115, pp. 64–68, 2007.

[11] M. H. Alalfi, J. R. Cordy, and T. R. Dean, "Analysis and clustering of model clones: An automotive industrial experience," in *CSMR-WCRE*, Feb 2014, pp. 375–378.

[12] M. Stephan and J. R. Cordy, "A survey of model comparison approaches and applications." in *International Conference on Model-Driven Engineering and Software Development*, 2013, pp. 265–277.

[13] F. Deissenboeck, B. Hummel, E. Juergens, M. Pfaehler, and B. Schaetz, "Model clone detection in practice," in *International Workshop on Software Clones*. ACM, 2010, pp. 57–64.

[14] M. H. Alalfi, T. R. Dean, J. R. Cordy, M. Stephan, and A. Stevenson, "Models are Code too: Near-miss Clone Detection for Simulink Models," in *ICSM*, 2012, pp. 295–304.

[15] H. Störrle, "Towards clone detection in UML domain models," *Software & Systems Modeling*, vol. 12, no. 2, pp. 307–329, 2013.

[16] N. Pham, H. Nguyen, T. Nguyen, J. Al-Kofahi, and T. Nguyen, "Complete and accurate clone detection in graph-based models," in *ICSE*, 2009, pp. 276–286.

[17] E. Antony, M. H. Alalfi, and J. R. Cordy, "An Approach to Clone Detection in Behavioural Models," in *International Working Conference in Reverse Engineering*, 2013, pp. 472–476.

[18] Ö. Babur, L. Cleophas, and M. van den Brand, "Metamodel clone detection with SAMOS," *Journal of Computer Languages*, 2019.

[19] D. Strüber, V. Acreţoaie, and J. Plöger, "Model clone detection for rule-based model transformation languages," *SoSyM*, pp. 1–22, 2017.

[20] M. Stephan and J. R. Cordy, "Identifying Instances of Model Design Patterns and Antipatterns Using Model Clone Detection," in *MISE*, May 2015, pp. 48–53.

[21] ——, "Identification of Simulink model antipattern instances using model clone detection," in *MODELS*, Sept 2015, pp. 276–285.

[22] M. H. Alalfi, E. P. Antony, and J. R. Cordy, "An approach to clone detection in sequence diagrams and its application to security analysis," *Software & Systems Modeling*, vol. 17, no. 4, pp. 1287–1309, Oct 2018.

[23] M. Stephan and J. R. Cordy, "Model-Driven Evaluation of Software Architecture Quality Using Model Clone Detection," in *QRS*, 2016, pp. 92–99.

[24] ——, "MuMonDE: A framework for evaluating model clone detectors using model mutation analysis," *Software Testing, Verification and Reliability*, vol. 21, no. 1-2, p. e1669, 2018.

[25] F. Ciccozzi, M. Famelis, G. Kappel, L. Lambers, S. Mosser, R. F. Paige, A. Pierantonio, A. Rensink, R. Salay, G. Taentzer, A. Vallecillo, and M. Wimmer, "Towards a body of knowledge for model-based software engineering," in *MODELS Companion Proc.* ACM, 2018, pp. 82–89.

[26] S. Koch, S. Strecker, and U. Frank, "Conceptual modelling as a new entry in the bazaar: The open model approach," in *IFIP International Conference on Open Source Systems*. Springer, 2006, pp. 9–20.

[27] R. France, J. Bieman, and B. H. Cheng, "Repository for model driven development (ReMoDD)," in *MODELS*. Springer, 2006, pp. 311–317.

[28] M. La Rosa, H. A. Reijers, W. M. Van Der Aalst, R. M. Dijkman, J. Mendling, M. Dumas, and L. García-Bañuelos, "Apromore: An advanced process model repository," *Expert Systems with Applications*, vol. 38, no. 6, pp. 7029–7040, 2011.

[29] S. E. Sim, S. Easterbrook, and R. C. Holt, "Using benchmarking to advance research: A challenge to software engineering," in *ICSE*, 2003, pp. 74–83.

[30] C. K. Roy and J. R. Cordy, "Benchmarks for software clone detection: A ten-year retrospective," in *SANER*. IEEE, 2018, pp. 26–37.

[31] S. Bellon, R. Koschke, G. Antoniol, J. Krinke, and E. Merlo, "Comparison and evaluation of clone detection tools," *IEEE Transactions on software engineering*, vol. 33, no. 9, 2007.

[32] J. Svajlenko and C. K. Roy, "Evaluating clone detection tools with bigclonebench," in *ICSME*, Sep. 2015, pp. 131–140.

[33] A. Charpentier, J.-R. Falleri, D. Lo, and L. Réveillère, "An empirical assessment of bellon's clone benchmark," in *EASE*, 2015, p. 20.

[34] F. Basciani, J. Di Rocco, D. Di Ruscio, L. Iovino, and A. Pierantonio, "Model repositories: Will they become reality?" in *CloudMDE@ MoDELS*, 2015, pp. 37–42.