Metamodel Clone Detection with SAMOS (extended abstract)

Önder Babur, Loek Cleophas, Mark van den Brand
Eindhoven University of Technology
Eindhoven, The Netherlands
{O.Babur, L.G.W.A.Cleophas, M.G.J.v.d.Brand}@tue.nl

I. Extended Abstract

Model-driven engineering (MDE) promotes the use of models (and metamodels to which they conform) as central artifacts in the software development process. This eases development and maintenance of software artifacts (including source code generated from models), yet increasing MDE adoption leads to an abundance of models in use. Some examples of this include the academic efforts to gather models in repositories, and large-scale MDE practices in the industry. This leads to challenges in the management and maintenance of those artifacts. One of those challenges is the identification of model clones, which can be defined in the most general sense as duplicate or highly similar models and model fragments. Similar scenarios apply in the traditional software development for source code clones. There is a significant volume of research on code clones, elaborating the drawbacks of having clones, which can be a major source of defects or lead to higher maintenance cost and less reusability, and providing detection techniques and tools. Note that in some cases clones might be useful too; it is nevertheless worthwhile to investigate them. Code clones have attracted the attention of the source code analysis community, who had to deal with the maintenance of large numbers of artifacts for a longer time than the MDE community.

Model clone detection, on the other hand, is a relatively new topic. Many researchers have drawn parallels to code clones, and claimed that a lot of the issues there can be directly translated into the world of models. While the problem domains are similar, the solution proves to be a challenge. Source code clone detection usually works on linear text or on an abstract syntax tree of the code, while models in general are graphs; other aspects are also inherently different for models, such as tool-specific representations, internal identifiers, and abstract vs. concrete syntaxes.

In our research we have the goal of detecting clones in large repositories of models (notably Ecore metamodels) and large evolving industrial domain-specific language (DSL) ecosystems based on the Eclipse Modelling Framework (EMF). Metamodels are artifacts of particular interest to us, for various purposes including metamodel repository management and DSL analysis. To achieve this goal, we have investigated the feasibility of existing tools, with three major requirements: (1) conceptual and technological applicability to Ecore metamodel clones; (2) sensitivity to all possible metamodel changes, and accuracy in general (precision, recall); and (3) scalability for large datasets. Among the existing model clone detectors in the literature are ConQAT, NICAD-SIMONE and MACH. A good portion of such tools are either limited to, tailored for, or evaluated on specific types of models such as MATLAB/Simulink. As a starting point we considered MACH and NICAD-SIMONE as promising candidates. However, these tools underperformed with respect to some of our requirements, notably sensitivity with respect to fine-grained changes, thus accuracy. We have eventually taken an orthogonal approach by extending the SAMOS framework (Statistical Analysis of MOdelS) for model clone detection. SAMOS is a state-of-the-art tool for large-scale analysis of models. We wish to exploit the underlying capabilities of the framework—incorporating information retrieval-based fragmentation, natural language processing, and statistical algorithms—for model clone detection. This extended abstract summarizes our recent studies ([1], [2]) in extending and tailoring SAMOS for (meta-)model clone detection, and evaluating it in three extensive case studies with mutation analysis for SAMOS, NICAD and MACH; comparison of SAMOS with NICAD on ATL Zoo metamodels; and finally a repository mining scenario on a very large set of GitHub metamodels.

SAMOS applies a typical Information Retrieval plus Machine Learning workflow (see Figure 1) to models: (1) extract a set of features such as model names, types, and chunks out of the underlying model graph (e.g. n-grams or subtrees), (2) define comparison and weighting schemes for those features such as Natural Language Processing (NLP) for model names and tree edit distance for subtrees, (3) compute a Vector Space Model (VSM) based on the feature comparison for all the models in the dataset, and finally (4) apply distance measures and clustering suitable to the problem at hand. In our study we have tailored this workflow for clone detection with additional scoping capabilities (e.g. whole model or EClass scope), a new distance measure (masked Bray-Curtis) and a density-based clustering algorithm to find metamodel clones.

We performed three case studies to evaluate the clone detection capabilities of SAMOS compared to NICAD and MACH; in terms of accuracy, and with respect to scalability in the presence of e.g. thousands of models:

- **Case Study 1:** We analyzed artificially generated atomic mutation cases and large change scenarios where we

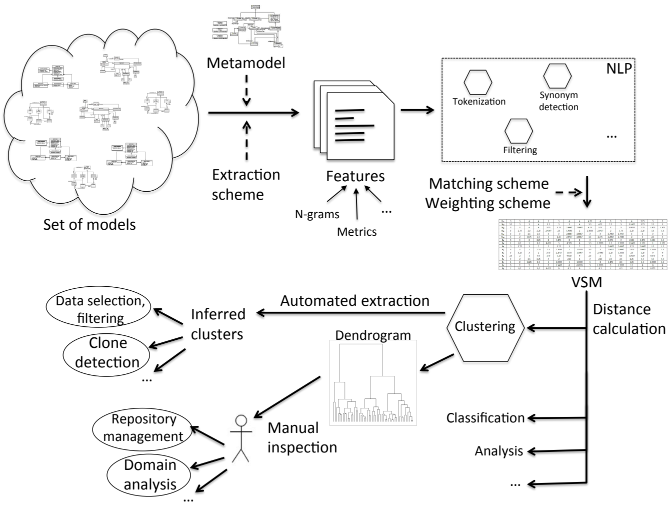

Fig. 1. Overview of SAMOS workflow.

measured pairwise distances with the base metamodel and the mutated ones to see how sensitive and accurate the different settings of SAMOS (unigrams, bigrams, subtrees with two comparison methods) are, compared to NICAD and MACH.

- **Case Study 2:** Comparing model fragments at EClass scope, we ran SAMOS with the most accurate setting along with NICAD on the configuration management metamodels from ATL Zoo[1], and comparatively evaluated the clone pairs and their correct classification, respective precision and recall, as found by the two tools.
- **Case Study 3:** We performed a large-scale clone detection exercise in two steps, with the cheapest and most expensive settings of SAMOS. We aimed to find the metamodel clone clusters in GitHub, for data preprocessing and filtering purposes, and with an eye towards future empirical studies on metamodels and domain specific languages (DSLs).

We obtained valuable findings in all three case studies. In case study 1, we found out that SAMOS performed well thanks to its NLP capabilities and fine-tuning, yet failed to detect certain types of move and swap mutations. NICAD on the other hand detected all cases except ones requiring NLP — which might occur frequently in real world data (as confirmed by our next case study), but over-approximated distances with its line-based approach. MACH on the other hand performed similar to the simpler configurations of SAMOS hence missed quite a few of the mutations. In case study 2, we compared SAMOS with its most accurate subtree setting to NICAD, on conference management metamodels from ATL Zoo. Mostly thanks to its NLP capabilities, SAMOS performed better in terms of both precision and recall, for Type A (content-wise duplicate), B (with very small changes) and C (highly similar) clones for that dataset (see Table I). We nevertheless identified certain problems with SAMOS' clone detection (some of which actually apply to NICAD and MACH as well), e.g. in terms of orthogonality of the changes among models, and connectivity clustering long strips of models. In the final case study, we turned to cluster nearly *all* the metamodels in GitHub, namely 68,511 items. After eliminating 2/3 of the dataset as exact file duplicates, we applied an iterative clone detection process. We first ran a cheap and inaccurate setting of SAMOS to pre-cluster the data into potential buckets, and did a more accurate pass on each bucket separately. As a result we found (with high precision, as we qualitatively evaluated) that still a high number of metamodel clone clusters can be found: content-wise (near-)duplicates (involving ∼8k models in total), or metamodels that are still highly similar (involving ∼11k models) for Type C). We regard this finding as a basis for our future studies on the one hand, and an important piece of information for anyone doing future empirical studies on those metamodels.

		Type A	Type B_{ex}	Type C_{ex}
SAMOS	precision	1.00	0.89	0.65
	rel. recall	1.00	0.92	0.78
NICAD	precision	1.00	0.46	0.26
	rel. recall	0.75	0.09	0.34

TABLE I
PRECISION AND RELATIVE RECALL OF SAMOS AND NICAD.

We have developed a novel model clone detection approach based on the SAMOS model analytics framework using information retrieval and machine learning techniques. We have extended SAMOS with additional scoping, feature extraction and comparison schemes, customized distance measures and clustering algorithms in the context of metamodel clone detection. We have evaluated our approach using a variety of case studies involving both synthetic and real data; and identified the strengths and weaknesses of our approach along with two other state-of-the-art clone detectors, namely NICAD and MACH. We conclude that SAMOS stands out with its higher accuracy yet considerable scalability for further large-scale clone detection and empirical studies on metamodels and domain specific languages. While SAMOS has many strengths, including its genericness with respect to the modelling languages, its plug-in architecture, and support for distributed computing, sophisticated NLP, and advanced statistical techniques using R, there are quite some directions for future work. We plan to improve SAMOS in terms of accuracy with additional features and comparison schemes, based on its weaknesses observed in the mutation analyses, customized and improved weighting schemes, NLP capabilities, distance measures and statistical algorithms. Furthermore we are working on a full-fledged distributed computing backend for SAMOS for further scalability. We also plan to tackle the extra-functional aspects of model clone detection such as clone ranking, reporting, inspection and visualization in future work.

[1] http://web.imt-atlantique.fr/x-info/atlanmod/index.php?title=Ecore

References

[1] Ö. Babur, “Clone Detection for Ecore Metamodels using N-grams,” Proceedings of the 6th International Conference on Model-Driven Engineering and Software Development, MODELSWARD 2018, pp. 411–419.

[2] Ö. Babur, L. Cleophas, M. van den Brand, “Metamodel Clone Detection with SAMOS,” Journal of Visual Languages and Computing (JVLC), Accepted with minor revision.