Predicting the Performance of ATL Model Transformations (extended abstract)

Raffaela Groner 1, Peter Bellmann 2, Stefan Höppner 1, Patrick Thiam 3, Friedhelm Schwenker 4 und Matthias Tichy 1

Abstract: Model transformations are used in different areas of computer science. They are often used during the development of software but also at runtime, e.g., to update a digital twin. However, users of these languages still suffer from very limited tool support, especially in the context of performance engineering techniques, users are left to their own devices. In particular, performance prediction approaches for model transformations are unknown until now. To tackle this issue, we performed several experiments in which we used three different machine learning approaches to predict the performance of five transformations written in the Atlas Transformation Language (ATL) based on different characterizations of input models. In our experiments, we trained a random forest, a linear regression, and a support vector regression model with a radial basis functional kernel using real-world models. The results of our experiments show that we can achieve a mean absolute percentage error between 2.07% and 10.63% using support vector regression and a straightforward model characterization consisting of the number of model elements, the number of references, and the number of attributes. We also show a weakness of our approach which fails to predict the performance of transforming attributes with an arbitrary size, like string attributes. More details about our experiments can be found in our corresponding publication, which was accepted by the 2023 ACM/SPEC International Conference on Performance Engineering (ICPE) [Gr23].

Keywords: performance prediction, ATL, model transformation, machine learning, linear regression, random forests, support vector regression

The input models that have to be transformed have a significant influence on the execution time of a model transformation. However, to analyze the performance large models are necessary but their manual creation is very time-consuming and error-prone, as models have to fulfill structural and semantic constraints. To get a first impression of the performance without creating a model, a what-if analysis for model transformations comes in handy that helps to analyze how a system responds to a described input. To realize such an analysis, we performed several experiments in which we trained different machine learning approaches to predict the execution time of six ATL transformations.
Our prediction approach uses a characterization of a model as input and provides a prediction based on this description. By iteratively combining structure and type information from Ecore models, we have defined eight different feature sets that can be used to characterize the input models. These feature sets consist, e.g., of the number of model elements per model element type. We compare the results of our eight individual feature sets with the results we obtain using a feature selection method on the union of our feature sets. As a selection method, we use variance thresholding, which removes features whose variance is below a predefined threshold. We use five different thresholds to filter the union of our eight feature sets. In our experiments, we compare three different machine learning approaches, namely linear regression (LR), random forest (RF), and support vector regression using a radial basis function kernel (SVR). We use six ATL transformations from the ATL Zoo, for which we use real-world input models. For each combination of transformation, feature set, and machine learning approach, we perform a 10-fold cross-validation evaluation. We also optimize the hyperparameters for the RF and SVR approaches. To evaluate our results, we use the mean absolute percentage error (MAPE) in %.

Our results show that the RF and the SVR approaches outperform the LR approach and that there exists no linear relationship between our feature sets and the execution time of transformations. The SVR approach in combination with feature selection based on variance thresholding yields the best results with a MAPE of 3.52% over all modules excluding one for which our approach fails. However, a better choice for a what-if analysis is the SVR approach in combination with another feature set, which describes a model based on the number of model elements, the number of references between the model elements, and the number of attributes. This feature set is much simpler than the union of our eight individual feature sets and the results are only slightly worse with a MAPE of 4.45%. Because our feature sets only contain structure and type information, our approach fails for transformations that transform very large string attributes. Since strings can have any size and affect performance, we focus our future work on how to include them as additional features.

This work was partially funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) - Ti 803/4-1. The authors acknowledge the support by the state of Baden-Württemberg through bwHPC.

Literatur