Visualizing Model and Data Differences with Inline Diff Editors in an Enterprise Low-Code Platform

Arvid Butting[1], Timo Greifenberg[1], Katrin Hölldobler[1], and Timo Kehrer[2]

Abstract: Computing the difference between models is a foundation for solving various challenges in model-driven engineering. Models of low-code platforms are often of non-textual nature to be suited best for citizen developers as their creators. This poses special challenges for visualizing model differences. In this talk, we extend an approach for visualizing model differences via diff editors derived from the original model editors of the low-code platform by a novel kind of diff representation called *inline diff editors*. The approach improves on the previous approach through a concise form of presentation and by avoiding the generation of new user interface and data models for the editors.

Keywords: Low-Code Platform, Model Differencing, A12

1 Presentation Description

In model-driven engineering, comparing models to compute their difference ("model differencing") is a foundation for solving various challenges, such as, establishing asynchronous collaborative editing of models, realizing model refactoring, and synthesizing proposals for repairing inconsistent models.

Low-code platforms employ models of various kinds that describe (parts of) a low-code application. For instance, data models typically describe the data structure of an application, user interface (UI) models model the elements of the user interface, and behavioral models describe business processes or UI flows. Model differencing has some particular challenges if applied to low-code software development. (1) Users of low-code platforms, often referred to as citizen developers, are not necessarily familiar with software development. Being forced to use software development tools, such as classical, line-based differencing tools that are typically part of integrated development environments (IDEs), would be an accidental complexity for citizen developers. Instead, a suitable form of presenting the differences is of utmost importance. (2) Low-code platforms use models in the application development platform and data conforming to these models in the low-code applications. Tooling to compute differences is useful on both levels. In model-driven engineering, a modeling tool that is developed using the same (meta-) modeling techniques that it can process is called bootstrapped. A bootstrapped low-code platform can employ the same approach for calculating model differences and data differences.

1 mgm technology partners gmbh, Aachen, Germany, Arvid.Butting@mgm-tp.com; Timo.Greifenberg@mgm-tp.com; Katrin.Hölldobler@mgm-tp.com
2 University of Bern, Bern, Switzerland, timo.kehrer@unibe.ch
This work is licensed under Creative Commons Attribution 4.0 International License http://creativecommons.org/licenses/by/4.0/, https://doi.org/10.18420/modellierung2024-ws-025

A12 [Me23] is an enterprise low-code application for developing large-scale, data-driven software applications. It focuses on customizability and exchangeability of its components to support custom software engineering and maintenance of long-lasting applications. Furthermore, it supports more than 10 different model types to describe applications. Among these, document models and relationship models are the basis for data modeling. Form models, tree models, and overview models describe the UI. The central modeling tool of A12 is the Simple Model Editor (SME) that contains a workspace explorer to manage the models of an application and individual editors for the different supported model types. The SME is bootstrapped and, hence, uses document and relationship models as meta models and form, overview, and tree models for describing the UI of the model editors and the workspace explorer.

In [Bu23], we described an approach for calculating structural model and data differences via traversing and matching trees of model elements. The approach integrates into enterprise low-code platforms and focuses on visualizing the differences by means of a delta overview and detailed diff editors. Beyond the calculation of the differences, it contains a mechanism to generate data and UI models for visualizing the difference in side-by-side diff editors in a notation close to the environment that is used for creating and editing models.

In this talk, we extended the approach in two dimensions. We present a novel approach for systematically deriving inline model diff editors to highlight model and data differences integrated with the unchanged model and data parts. The inline model diff editors reuse the original editor models. Therefore, it is not necessary to generate diff editor models anymore. Instead, the UI model interpreters are extended by customized UI components (called widgets) to display the inline model diff in the user interface. These customized widgets highlight added, removed, and modified model elements both by color-coding and, for accessibility reasons by means of a short textual description. We extended the original approach for triggering the difference calculation. Now it is possible to compare different models of the workspace directly to each other. By this, the approach already supports a large number of applications in which variants or co-supported versions of the same model are maintained in a single workspace. Additionally, we support asynchronous collaborative modeling based on Git. In this, we enable comparing models of different versions by browsing the Git-based model history. With the two extensions presented in this talk, the approach covers a wider range of use cases for triggering model differencing and improves the efficiency of the approach, because no intermediate model generation is required anymore.

References

[Bu23] Butting, A.; Greifenberg, T.; Hölldobler, K.; Kehrer, T.: Model and Data Differences in an Enterprise Low-Code Platform. In: 2023 ACM/IEEE International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C). IEEE, pp. 868–877, 2023.

[Me23] Mehmanesh, H.; Gandenberger, S.; Weiss, A.; Kneist, T.; Lorenz, S.; Backschat, M.: A12 – Low Code for Custom Enterprise Software, tech. rep., mgm technology partners GmbH, 2023, URL: https://www.mgm-tp.com/documents/A12_Whitepaper_EN.pdf.