

# Assessing the Usefulness of a Visual Programming IDE for Large-Scale Automation Software

Bianca Wiesmayr,<sup>1</sup> Alois Zoitl<sup>2</sup> Rick Rabiser<sup>3</sup>

**Abstract:** This is a summary of a paper (with the same title) that we published at the ACM/IEEE 24th International Conference on Model-Driven Engineering Languages and Systems (MODELS 2021) describing a study centered on a visual programming IDE for large-scale automation software development and maintenance.

**Keywords:** Usefulness study; open source software; IEC 61499; modeling tools; model-driven engineering

## 1 Summary

Industrial control software is more commonly developed and maintained by domain experts than by software engineers. These control engineers frequently use visual programming languages based on industrial standards such as IEC 61131-3 and IEC 61499. The latter standard applies model-based engineering concepts to abstract from hardware and low-level communication. Industrial control systems are usually complex, one-of-a-kind systems that are maintained for many years. Therefore, very usable IDEs for visual programming languages are required. However, not much empirical research exists on the practical usefulness of such IDEs, i.e., usability and utility.

We have conducted a multi-stage usability study to evaluate the usefulness of an open source modeling tool for IEC 61499, the 4diac IDE [Ec21]. In our paper [WZR21a], we discuss common control software maintenance tasks and tool capabilities derived from existing research and show their realization in the 4diac IDE. In the first stage, we performed a walkthrough of the capabilities using the cognitive dimensions of notations framework, where developers evaluated the tool usability. After improving the tool, we conducted a remote user study with ten industrial automation engineers. We asked each participant to perform maintenance tasks on provided automation software for a capping station. The study leader instructed the participant, while one scribe collected think-aloud statements and another scribe noted observations. After the practical work with the tool, we conducted a semi-structured utility interview and asked the subjects to fill in a usability questionnaire.

---

<sup>1</sup> LIT CPS Lab, Johannes Kepler University Linz, Altenberger Str. 69, 4040 Linz, Austria, bianca.wiesmayr@jku.at

<sup>2</sup> CDL VaSiCS, LIT CPS Lab, Johannes Kepler University Linz, Austria, alois.zoitl@jku.at

<sup>3</sup> CDL VaSiCS, LIT CPS Lab, Johannes Kepler University Linz, Austria, rick.rabiser@jku.at

We grouped all results based on the cognitive dimensions. As expected, the understandability and maintainability of models depends on the provided IDE features. For example, we learned that the layout quality affects understanding visual models, but manually adjusting the graph layout is tedious. Users therefore benefit from an automated layout. However, they also need capabilities for customizing the layout algorithms to match their mental model. Orienting in a large application could be further facilitated with search features and reference lists. Finally, maintaining existing software may require complex editing operations. A modeling IDE should support these operations by handling inconsistencies gracefully. In general, we conclude that advanced tools with sophisticated editing support can simplify working with large models and thus increase the benefits of the applied modeling language.

Our findings demonstrate how the usefulness of IDEs can be successfully investigated using a multi-phase approach combining a walkthrough and a user study. Our results and lessons learnt from the study are relevant for developers of visual modeling and programming tools. The identified capabilities for software maintenance are often not sufficiently supported in such tools. Furthermore, our results with the Eclipse-based modeling tool 4diac IDE are potentially applicable to other modeling tools using the same technology. As our improvements are included in the latest open-source release of the 4diac IDE, they can serve as a good practice example for other project teams, together with this paper.

## 2 Data Availability

The tool 4diac IDE is part of the Eclipse 4diac<sup>TM</sup> project and therefore available open source, including all extensions that were developed as part of the presented study. Furthermore, also all documents from the study (interview questions, questionnaire, study system) are available online [WZR21b].

## Literatur

- [Ec21] Eclipse 4diac: Eclipse 4diac - The Open Source Environment for Distributed Industrial Automation and Control Systems, 2021, URL: [www.eclipse.org/4diac](http://www.eclipse.org/4diac).
- [WZR21a] Wiesmayr, B.; Zoitl, A.; Rabiser, R.: Assessing the Usefulness of a Visual Programming IDE for Large-Scale Automation Software. In: ACM/IEEE 24th International Conference on Model Driven Engineering Languages and Systems (MODELS). IEEE, 2021.
- [WZR21b] Wiesmayr, B.; Zoitl, A.; Rabiser, R.: Assessing the Usefulness of a Visual Programming IDE for Large-Scale Automation Software: Documents for performing a user study on maintaining hierarchical control applications with 4diac IDE. 2021, URL: [doi.org/10.5281/ZENODO.4758816](https://doi.org/10.5281/ZENODO.4758816).