EDD: A Web-Based Editor for Declarative Process Models Using easyDeclare

Graziano Blasilli[1,†], Lauren S. Ferro[2,†], Simone Lenti[1,†], Fabrizio Maria Maggi[2,†], Andrea Marrella[1,*,†] and Tiziana Catarci[1,†]

[1] *Sapienza University of Rome, Roma, Italy*

[2] *CSIRO's Data61, Clayton, Australia*

[3] *Free University of Bozen-Bolzano, Bolzano, Italy*

Abstract

In process mining, declarative process discovery aims to extract declarative process models from event logs. Declare is a declarative process modeling language that provides an extensible set of standard templates for modeling declarative processes imposing temporal constraints on the sequence of process activities. The easyDeclare graphical notation has recently been developed to represent Declare models. This graphical notation has been proven to enhance human understandability of declarative process discovery results and reduce the cognitive load required to interpret these models compared to the original Declare graphical notation. This paper presents EDD, a web-based editor for declarative process models using the easyDeclare notation. Researchers and practitioners in the process mining field have evaluated the usability of EDD, resulting in a high usability rating. The tool can be used directly within a web browser or by cloning its GitHub repository. As an open-source project, EDD is available to the research community for use and further improvement.

Keywords

Declare, easyDeclare, Declarative Process Models, Declarative Process Discovery, Visualization, Editor

Metadata description	Value
Tool name	EDD
Current version	1.0
Legal code license	MIT
Languages, tools and services used	JavaScript
Supported operating environment	All
Download/Demo URL	https://blasilli.github.io/EDD
Documentation URL	https://github.com/blasilli/EDD
Source code repository	https://github.com/blasilli/EDD
Screencast video	https://blasilli.github.io/EDD/tutorial.mp4

ICPM Doctoral Consortium and Demo Track 2024, October 14-18, 2024, Kongens Lyngby, Denmark

[*] Corresponding author.

[†] These authors contributed equally.

blasilli@diag.uniroma1.it (G. Blasilli); lauren.ferro@data61.csiro.au (L. S. Ferro); lenti@diag.uniroma1.it (S. Lenti); maggi@inf.unibz.it (F. M. Maggi); marrella@diag.uniroma1.it (A. Marrella); catarci@diag.uniroma1.it (T. Catarci)

0000-0003-3339-6403 (G. Blasilli); 0000-0003-4064-8218 (L. S. Ferro); 0000-0001-8281-3723 (S. Lenti); 0000-0002-9089-6896 (F. M. Maggi); 0000-0002-1031-0374 (A. Marrella); 0000-0002-3578-1121 (T. Catarci)

1. Introduction

Declarative process models capture the behavior of a business process by imposing temporal constraints on the sequence of process activities. In process mining, declarative process discovery aims to extract these constraints from event logs. Declare is a declarative process modeling language [1] that specifies an extensible set of standard templates that a process analyst can use to model a process. Declare has proven beneficial in modeling flexible processes where it is hard to specify in advance a prescriptive order of process activities to be executed, particularly in healthcare [2, 3] and emergency management scenarios [4, 5, 6].

Constraints are concrete instantiations of Declare templates. Declare is equipped with formal semantics based on Linear Temporal Logic on Finite Traces [7], but the use of templates makes model comprehension independent of the logic-based formalization. Analysts can work with the graphical representation of templates while the underlying formulas remain hidden. Graphically, a Declare model is a diagram in which activities are represented as nodes (labeled rectangles) and constraints as arcs between activities.

Although the semantic aspects of declarative processes have been extensively investigated [8], there has been less focus on designing declarative visual notations that enhance model understanding and assist analysts in performing process mining tasks. To improve the human understandability of declarative process models, in [9] we introduced easyDeclare, a novel visual notation for declarative process modeling, developed with consideration of the well-established Moody's design principles [10] and designed to improve the human understandability of declarative models based on the Declare language. This visual notation has been proven to reduce the cognitive load required to interpret Declare models compared to the original Declare notation, making it a promising alternative to enhancing overall comprehension of the outcomes of declarative process discovery tasks.

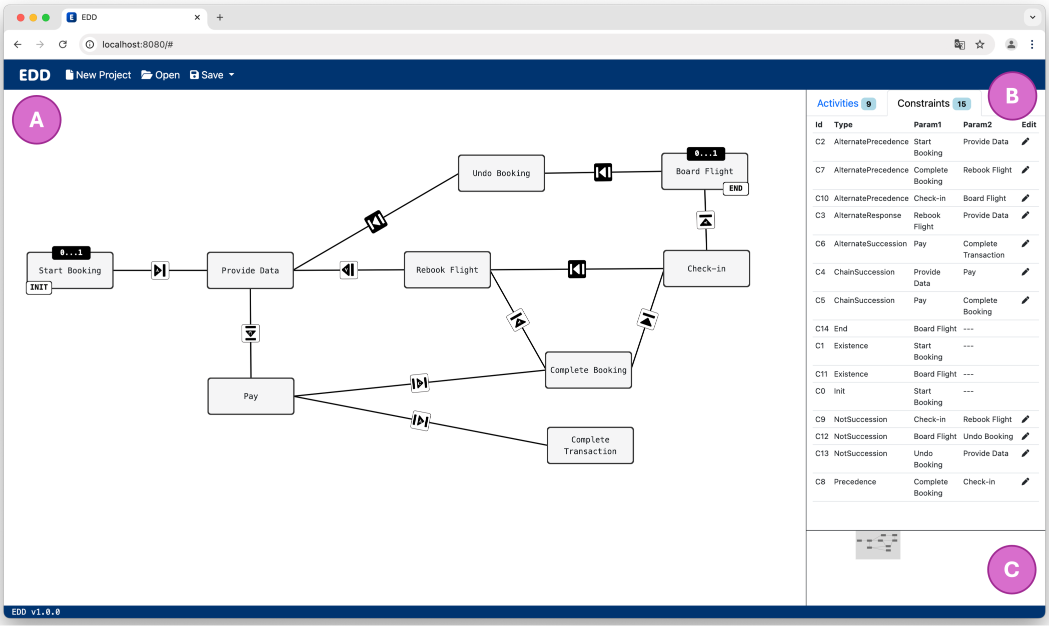
This paper presents EDD (easyDeclare Designer), an open-source web application developed in JavaScript. EDD is the first editor that allows working with declarative process models based on the novel easyDeclare graphical notation. To assess its usability, we conducted a user study involving researchers and practitioners from the process mining field, using the System Usability Scale (SUS) questionnaire [11]. The results of this evaluation were highly positive, with the tool receiving an average grade of A+, indicating that users found EDD to have a high level of usability.

The rest of the paper is organized as follows. Section 2 presents the visual components and features offered by EDD to model declarative processes using easyDeclare. Section 3 discusses the results of a user study conducted to assess the tool's usability, and discusses the maturity level of the tool. Finally, Section 4 concludes the paper.

2. The EDD Tool

EDD is an editor for declarative process models using the easyDeclare visual notation. We developed EDD as an open-source web application implemented in JavaScript, and usable with any browser. It can either be used directly in the browser [1] or by cloning the GitHub repository.

[1] https://blasilli.github.io/EDD



Id	Type	Param1	Param2	Edit
C2	AlternatePrecedence	Start Booking	Provide Data	✎
C7	AlternatePrecedence	Complete Booking	Rebook Flight	✎
C10	AlternatePrecedence	Check-in	Board Flight	✎
C3	AlternateResponse	Rebook Flight	Provide Data	✎
C6	AlternateSuccession	Pay	Complete Transaction	✎
C4	ChainSuccession	Provide Data	Pay	✎
C5	ChainSuccession	Pay	Complete Booking	✎
C14	End	Board Flight	---	
C1	Existence	Start Booking	---	
C11	Existence	Board Flight	---	
C0	Init	Start Booking	---	
C9	NotSuccession	Check-in	Rebook Flight	✎
C12	NotSuccession	Board Flight	Undo Booking	✎
C13	NotSuccession	Undo Booking	Provide Data	✎
C8	Precedence	Complete Booking	Check-in	✎

Figure 1: Overview of the EDD interface. The *Canvas Pane* (A) shows the model and offers an interactive interface for model creation and manipulation. The *Details Pane* (B) provides an overview of the existing activities and constraints in the model, allowing the editing of their properties. The *Overview Pane* (C) shows the overview of the entire model following the focus+context visual paradigm.

User Interface The user interface has been designed to be simple and efficient. As shown in fig. 1, it is composed of three panels: A) *Canvas Pane*, B) *Details Pane*, and C) *Overview Pane*. The main pane is the *Canvas* (A) on the left side of the interface, which shows the model. This pane also offers an interactive interface for model creation and manipulation. Users can create activities by double-clicking on an empty area of the canvas. To create binary constraints between activities, users can mouse over the border of an activity and drag the appearing arrow to another activity. Although the tool automatically arranges activities and constraints on the canvas to maintain an organized layout, users can manually adjust the positions by dragging these elements across the canvas, allowing for a custom layout. Furthermore, the canvas supports zoom and pan functionalities, allowing users to manage models of various sizes. On the right side of the interface is the *Details Pane* (B), which provides an overview of the existing activities and constraints in the model. This pane also shows details about each activity and constraint and allows the editing of their properties. In the bottom right corner of the interface, there is the *Overview Pane* (C). This pane shows the overview of the entire model, highlighting which part is currently visible in the canvas, following the focus+context visual paradigm [12]. This feature helps users maintain a view of the entire model even if they zoom in and out on the canvas to work on different areas.

Templates EDD supports all the main Declare templates for constraints: 1) *Init* 2) *End* 3) *Absence* 4) *Existence* 5) *Choice* 6) *ExclusiveChoice* 7) *Response* 8) *Precedence* 9) *AlternateResponse*

10) *AlternatePrecedence* 11) *ChainResponse* 12) *ChainPrecedence* 13) *RespondedExistence* 14) *Co-Existence* 15) *NotCoExistence* 16) *Succession* 17) *NotSuccession* 18) *ChainSuccession* 19) *NotChain-Succession* 20) *AlternateSuccession.*

Import and Export To ensure interoperability with other tools, EDD supports the import and export of models in the "*decl*" file format [13] since it is widely used in the process mining community. In addition, we have also defined a new format called "*edj*". This format is a JSON-based file format specifically designed to include not only the list of activities and constraints but also the visual layout of the model. This allows users to preserve the positioning of elements within their models, making it easier to maintain consistency and clarity when models are shared or revisited. To be more interoperable, we also plan to use the *RuM* file format [14] in future versions of the tool.

3. Usability Evaluation

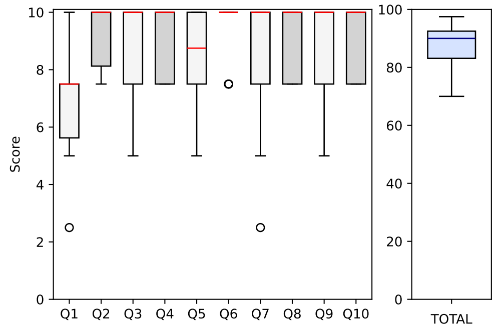
To evaluate the usability of EDD, we conducted a user study using the System Usability Scale (SUS) [11], a questionnaire designed to assess the perceived usability of a system. The user, after having performed tasks on the system, is required to answer to ten questions. These responses are then converted into numerical values and added together to obtain the total SUS score for the system which ranges from 0 to 100. This total score can be further mapped to 11 grades [15] from higher to lower: [*A+, A, A-, B+, B, B-, C+, C, C-, D, F*].

A total of 18 users were involved in the user study, both researchers and practitioners in the process mining field. The group is composed of 1 associate professor, 3 assistant professors, 10 PhD students, 2 master students, 1 bachelor student, and 1 process mining consultant.

Methodology After some profiling questions to the participants, we briefly recapped the basic concepts of Declarative Process Modeling and illustrated the easyDeclare visual notation. Then we presented EDDexplaining its functionalities. After, we allowed the participants to familiarize with the tool for ten minutes. To test the system functionalities, we designed two tasks (T1 and T2) for the users to complete. The first task has been designed to allow users to familiarize with the system by creating a simple model from scratch. The second task involved the editing of an existing and more complex model, which represents a booking procedure for flight tickets. After completing the tasks, participants were asked to fill out the SUS questionnaire. Finally, they were invited to share their comments and thoughts on the tool.

Results The SUS results show a highly positive evaluation of the tool usability, with an average grade of *A+* (87.64) and a median of 90. The standard deviation of $\sigma = 7.54$ suggests relatively low variability in scores, meaning that most users had a similar usability experience with EDD. These values indicate that users found the system very usable. The detailed results are shown in fig. 2.

Feedback Participants provided several positive comments and valuable suggestions for improving the tool. Users generally appreciated its visual design and usability. For example, one



Participant	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	TOTAL	GRADE
P1	7.5	7.5	7.5	7.5	10.0	7.5	7.5	7.5	10.0	7.5	80.0	A-
P2	5.0	10.0	7.5	7.5	7.5	10.0	5.0	7.5	5.0	10.0	75.0	B
P3	7.5	7.5	7.5	7.5	7.5	7.5	2.5	10.0	5.0	7.5	70.0	C
P4	10.0	10.0	10.0	7.5	7.5	10.0	10.0	10.0	10.0	10.0	95.0	A+
P5	7.5	10.0	10.0	10.0	7.5	10.0	10.0	10.0	10.0	10.0	95.0	A+
P6	5.0	10.0	10.0	7.5	10.0	10.0	10.0	10.0	10.0	10.0	92.5	A+
P7	10.0	10.0	7.5	10.0	7.5	10.0	7.5	10.0	10.0	7.5	90.0	A+
P8	7.5	10.0	10.0	10.0	7.5	10.0	7.5	10.0	10.0	7.5	90.0	A+
P9	5.0	7.5	10.0	7.5	7.5	10.0	10.0	7.5	7.5	10.0	82.5	A
P10	5.0	10.0	5.0	10.0	5.0	10.0	7.5	10.0	10.0	10.0	82.5	A
P11	7.5	7.5	10.0	7.5	10.0	10.0	10.0	7.5	7.5	7.5	85.0	A+
P12	7.5	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	97.5	A+
P13	2.5	10.0	7.5	10.0	7.5	10.0	10.0	10.0	7.5	10.0	85.0	A+
P14	10.0	10.0	10.0	10.0	10.0	7.5	10.0	10.0	10.0	10.0	97.5	A+
P15	7.5	10.0	10.0	7.5	10.0	10.0	10.0	7.5	10.0	10.0	92.5	A+
P16	7.5	10.0	7.5	10.0	10.0	10.0	7.5	10.0	7.5	10.0	90.0	A+
P17	7.5	10.0	10.0	10.0	10.0	7.5	10.0	7.5	10.0	7.5	90.0	A+
P18	7.5	7.5	10.0	10.0	10.0	10.0	7.5	7.5	7.5	10.0	87.5	A+
										AVERAGE	**87.64**	A+

Figure 2: System Usability Scale (SUS) results for EDD. These results indicate a high usability level, with an average grade of A+ (score 87.64).

user commented, "*Very nice visual tool, with clear feedback and high usability level*". Another noted, "*In general, I found the tool very simple to use*". Despite these positive remarks, users also offered constructive feedback for enhancements. One suggestion was to "*add support for multiple projects open at the same time in different tabs*", which would improve multitasking capabilities. Another user suggested, "*It would be helpful to double-click on an activity to rename it*", indicating a potential improvement in interaction design. Additionally, users also recommended keyboard shortcuts, such as "*clicking on an activity and pressing the delete key to remove the activity*".

Maturity and Development EDD was designed with a simple and clean interface to ensure ease of use, making it accessible even to users who may not be familiar with complex modeling tools. The tool currently offers minimal functionalities, focusing on core features that allow users to create and manage models effectively. Even if the users (through the SUS) have positively evaluated the offered functionalities, there is significant potential for further development, with opportunities to enhance its capabilities and improve the user experience. For instance, one possible enhancement could be the inclusion of graph layout algorithms to arrange activities and constraints in a more organized and visually appealing way. Another planned feature is the ability to export models as images, supporting both SVG and PNG formats.Since we released EDD as an open-source project, contributions from the community are highly encouraged, and any development efforts to extend its functionalities are greatly appreciated.

4. Conclusion

This paper presented EDD, an editor for declarative process models, using the easyDeclare visual notation. Through a comprehensive usability evaluation involving researchers and practitioners from the process mining field, EDD has demonstrated a high level of usability, as evidenced by its average SUS grade of A+. Even if the offered functionalities have been positively evaluated, there is still space for further development, with opportunities to enhance its capabilities and improve the user experience. For this reason, EDD has been released an open-source project; contributions from the community are highly encouraged and appreciated.

Acknowledgments. This work has been supported by the H2020 project DataCloud (Grant number 101016835), the Sapienza projects DISPIPE and FOND-AIBPM, and the PNRR MUR project PE0000013-FAIR.

References

[1] W. M. P. van der Aalst, M. Pesic, H. Schonenberg, Declarative workflows: Balancing between flexibility and support, Computer Science - R&D 23 (2009) 99–113.

[2] M. Rovani, F. M. Maggi, M. De Leoni, W. M. Van Der Aalst, Declarative process mining in healthcare, Expert Systems with Applications 42 (2015) 9236–9251.

[3] S. Agostinelli, F. Covino, G. D'Agnese, C. De Crea, F. Leotta, A. Marrella, Supporting governance in healthcare through process mining: a case study, Ieee Access 8 (2020) 186012–186025.

[4] S. R. Humayoun, T. Catarci, M. de Leoni, A. Marrella, M. Mecella, M. Bortenschlager, R. Steinmann, The WORKPAD User Interface and Methodology: Developing Smart and Effective Mobile Applications for Emergency Operators, in: Universal Access in Human-Computer Interaction. Applications and Services, 2009, pp. 343–352.

[5] S. R. Humayoun, T. Catarci, M. de Leoni, A. Marrella, M. Mecella, M. Bortenschlager, R. Steinmann, Designing mobile systems in highly dynamic scenarios: The WORKPAD methodology, Knowledge, Technology & Policy 22 (2009) 25–43.

[6] A. Marrella, M. Mecella, A. Russo, Collaboration on-the-field: Suggestions and beyond, in: 8th Int. Conf. on Information Systems for Crisis Response and Management, ISCRAM 2011, 2011.

[7] G. De Giacomo, M. Y. Vardi, Linear temporal logic and linear dynamic logic on finite traces, in: Twenty-Third Int. Joint Conf. on Artificial Intelligence (IJCAI), 2013, pp. 854–860.

[8] H. A. Lopez A., V. Simon, How to (re)design declarative process notations? A view from the lens of cognitive effectiveness frameworks, in: 15th IFIP Working Conference on the Practice of Enterprise Modeling 2022 (PoEM-Forum 2022), 2022.

[9] G. Blasilli, L. Ferro, S. Lenti, F. M. Maggi, A. Marrella, T. Catarci, Improving the understandability of declarative process discovery results using easydeclare, Preprint submitted at Information Systems (SI on Process mining meets visual Analytics) (2024). doi:10.5281/zenodo.13622518.

[10] D. Moody, The "physics" of notations: Toward a scientific basis for constructing visual notations in software engineering, IEEE Trans. on Software Engineering 35 (2009) 756–779.

[11] J. Brooke, SUS: a retrospective, Journal of usability studies 8 (2013) 29–40.

[12] H. Hauser, Generalizing focus+context visualization, in: Scientific Visualization: The Visual Extraction of Knowledge from Data, 2006, pp. 305–327.

[13] V. Skydanienko, C. Di Francescomarino, C. Ghidini, F. M. Maggi, A tool for generating event logs from multi-perspective declare models, volume 2196 of *CEUR WS Proc.*, 2018.

[14] A. Alman, C. Di Ciccio, D. Haas, F. M. Maggi, A. Nolte, Rule Mining with RuM, in: 2020 2nd Int. Conf. on Process Mining (ICPM), 2020, pp. 121–128.

[15] J. R. Lewis, J. Sauro, Item benchmarks for the system usability scale., Journal of Usability Studies 13 (2018).