ModeLLer – a Prototype to Support Requirements Elicitation in Co-design Environments

1st Chiara Mannari
National Research Council (CNR), University of Pisa
Pisa, Italy
chiara.mannari@isti.cnr.it

2nd Elisa Anichini
University of Pisa
Pisa, Italy
e.anichini1@studenti.unipi.it

3rd Manlio Bacco
National Research Council (CNR)
Pisa, Italy
manlio.bacco@isti.cnr.it

4th Alessio Ferrari
National Research Council (CNR)
Pisa, Italy
alessio.ferrari@isti.cnr.it

5th Tommaso Turchi
University of Pisa
Pisa, Italy
tommaso.turchi@unipi.it

6th Alessio Malizia
University of Pisa, Molde University College
Pisa, Italy and Molde, Norway
alessio.malizia@unipi.it

***Abstract*—This contribution presents ModeLLer, a prototype of a web tool for system modelling based on a block-based visual editor. The aim of ModeLLer is to enable collaborative environments in requirements elicitation, allowing end-users to create UML class diagrams without any knowledge of the (semi-)formal UML notation.**

***Index Terms*—process modelling, no-code, end-user development, socio-technical systems, requirements elicitation**

I. Context and Motivation

Requirements elicitation in socio-technical systems is carried out in collaborative environments involving interdisciplinary teams. Model-driven requirements engineering techniques (MoDRE) leverage diagrammatic notations as a means for providing easy-to-read representations of systems [1]. In a previous study [2], we proposed applying MoDRE techniques for the representation of systems to support the analysis of the impact of digitalisation in socio-technical contexts. The proposed procedure for system modelling is based on the elicitation of knowledge through interviews with domain experts, followed by a phase in which engineers formalise the system through visual diagrams, which are then illustrated to stakeholders for validation. Though this procedure facilitates elicitation of system-relevant information [4], [5], the usage of interviews may lead to phenomena of ambiguity and tacit knowledge that can only partially be addressed in the validation phase [6]. Furthermore, according to NAPIRE survey[1] "Communication flaws between the project and the customer" is perceived as the second most common problem in requirements engineering [7]. In this study, we introduce a tool to enable end-users to model systems without any knowledge of formal notations, and without the intervention of system engineers in the initial modelling phase. The tool is intended to speed up the process of system requirements and domain knowledge elicitation, solve information exchange issues, and provide a means to allow end-users to model systems through a no-code approach.

[1]Source: http://re-survey.org/#/explore, accessed 9 June 2023.

II. ModeLLer Prototype

We developed ModeLLer, a prototype of a web tool accessible to end-users for modelling systems[2]. The tool's current version enables users with no expertise in formal notations to generate UML class diagrams [8] by interacting with a block-based visual editor. The interface is divided into three main areas: a toolbox panel for the exploration and selection of blocks; a workspace where blocks can be added to and interlocked with other blocks to form the structure of the system; and an export area with export data and buttons (Fig.1). Models are created following a guided procedure prompted on the blocks. The preset block appearing on the initial setup of the workspace requires the identification of inner blocks which correspond to UML classes, i.e., actors and resources involved in the system. Inner blocks contain further suggestions requiring other interlocks or specifications, e.g. operations, attributes, and relationships. Complex connections as generalisation are also handled by the system at the block level. All the items dropped on the workspace are simultaneously translated into UML entities using the XMI format [9]. A textual report is also produced to support the user by providing a description of the model being created. The textual report also contains additional information for experts, such as notes and the motivation for activities carried out; in fact, the user is asked to provide information on activities according to the following pattern: an ⟨actor⟩ performs an ⟨activity⟩ using ⟨resource⟩ to ⟨motivation⟩.

Once the user has completed the creation of the model, it is possible to download the XMI file and the textual report. The code can be then imported into any UML design software supporting XMI[3]. A syntactically correct class diagram can be generated and further refined by experts selecting the XMI imported classes, which maintain features and relationships as

[2]ModeLLer is available on GitHub: https://github.com/Unipisa/blockly-UML-modeller

[3]We tested the XMI import in StarUML v.5.1.0 – https://staruml.io and Visual Paradigm 17.0 https://www.visual-paradigm.com, last accessed 8 June 2023

defined by the user in ModeLLer. The tool is a single-page application providing a visual block-based language based on the Blockly JavaScript library by Google [10]. Additional custom blocks for actors, resources, activities, etc., were created to provide a guided procedure in support of the users (Fig:2). The integration of custom blocks in Blockly consisted of two phases: the design of the blocks with custom styles definition (i.e., colour, texts, etc.), and the development of the blocks with the extension of the JSON source. The output component was also extended with a generating function to translate the custom blocks into XMI code and produce the textual report. An example of a simple model is represented in Fig. 3.

III. Evaluation

The prototype was evaluated through the cognitive walkthrough approach [11], [12]. This technique allowed us to evaluate the learnability and usability of ModeLLer from the user's point of view. A remote workshop was carried out with cross-functional evaluators with the aim of identifying aspects of the interface that could be difficult for users. The evaluators, three males and a female, aged between 35 and 45, are experts in the following areas: formal languages, visual languages, web design, and agricultural technology. Guided by a facilitator, the review team walked through each task of the modelling activity and answered a set of predetermined questions. Overall, the experts agreed on the positive evaluation of the tool. Table I summarises the actions analysed and the answers provided by the experts for each question. A redesign followed the workshop with the aim to solve the main issue identified by the evaluation team. i.e., the difficulty in the *creation of advanced structures*. This concerns the definition of associations between classes, which was evaluated as difficult, because the user was asked to create the association by typing in a text field the name of an already defined block. The issue was solved by replacing the text field with a selector in order to provide a list of possible relations with existing blocks and limit the user's errors.

IV. Discussion

The prototype can be adopted for requirements elicitation in co-design environments and can be tested in different scenarios. A potential scenario is the support of the classical methodology based on interviews to speed up process modelling and foster active participation. In such a case, the expert and the different stakeholders can co-create the diagrams, thus reaching a common understanding of the system. The approach is similar to the one adopted by PlantUML [13], but PlantUML is devoted to UML experts, while ModeLLer is based on a block language that can be understood by users without any knowledge of formal notations. Another potential scenario is the autonomous creation of models by end-users. This approach can be adopted in co-design environments requiring the participation of different stakeholders. In such scenarios, different models created from multiple perspectives can contribute to producing a holistic view of the system.

V. Questions for the Audience

The audience is encouraged to answer the following three questions: 1) *To what extent stakeholders outside the system engineering field would be willing to use the tool?* **Goal:** understand whether some issues may prevent users from accepting the tool in a requirements elicitation context. 2) *What are the main usability issues of the tool?* **Goal:** improve the tool's appeal and possibly facilitate its acceptance. 3) *To what extent does the tool facilitate the work of system engineers?* **Goal:** facilitate the acceptance of the tool by stakeholders and system engineers/requirements analysts involved in the elicitation process.

References

[1] M. Dumas , M. La Rosa , J. Mendling and H. A. Reijers, "Fundamentals of Business Process Management". Springer, 2018.

[2] C. Mannari, G. O. Spagnolo, M. Bacco and A. Malizia "Digitalisation of Agriculture: Development and Evaluation of a Model-based Requirements Engineering Process", REFSQ 2023 - 29th International Working Conference on Requirement Engineering: Foundation for Software Quality: Posters and Tools, Barcelona, Spain, 17-20/04/2023, 2023.

[3] S. Leminen, M. Westerlund, and AG. Nyström, "Living labs as open-innovation networks", 2012.

[4] G. Fischer and E. Giaccardi, "Meta-design: A framework for the future of end-user development", in End user development, 2006, pp. 427-457.

[5] H. Lieberman, F. Paternò, M. Klann and V. Wulf, "End-User Development: An Emerging Paradigm", End User Development, 2006.

[6] A. Ferrari, P. Spoletini and S. Gnesi, "Ambiguity and tacit knowledge in requirements elicitation interviews". Requirements Engineering 21, 2016.

[7] D.M. Fernández, S. Wagner, M. Kalinowski, M. Felderer, P. Mafra, A. Vetrò, T. Conte, M.-T. Christiansson, D. Greer, C. Lassenius, T. Männistö, M. Nayabi, M. Oivo, B. Penzenstadler, D. Pfahl, R. Prikladnicki, G. Ruhe, A. Schekelmann, S. Sen, R. Spinola, A. Tuzcu, J. L. de la Vara and R. Wieringa "Naming the pain in requirements engineering". Empirical Software Engineering 22, 2017.

[8] "Unified Modeling Language (UML) 2.5.1 Core Specification", Version 2. https://www.omg.org/spec/UML, 2017 — Online; accessed 9 June 2023.

[9] "XML Metadata Interchange (XMI)", Version 2.5.1. https://www.omg.org/spec/XMI, 2015 — Online; accessed 9 June 2023.

[10] Google, "Blockly", https://developers.google.com/blockly?hl=it, 2022 — Online; accessed 9 June 2023.

[11] Nielsen Norman Group, "Evaluate interface learnability with cognitive walkthroughs", https://www.nngroup.com/articles/cognitive-walkthroughs/, 2022 — Online; accessed 9 June 2023.

[12] Nielsen Norman Group, "How to conduct a cognitive walkthrough workshop". https://www.nngroup.com/articles/cognitive-walkthrough-workshop, 2022 — Online; accessed 4 May 2023.

[13] "PlantUML", https://plantuml.com — Online; accessed 9 June 2023.

Acknowledgment

This work has received funding from the EU Horizon Europe programme under GA no. 101060179.

APPENDIX

- A video demo of ModeLLer prototype is available at: https://github.com/Unipisa/blockly-UML-modeller/assets/76447624/fbc104ef-7100-4d1d-91f7-856071e24a4d
- The source code and a working demo of ModeLLer are available on Gihub: https://github.com/Unipisa/blockly-UML-modeller (accessed 9 June 2023)

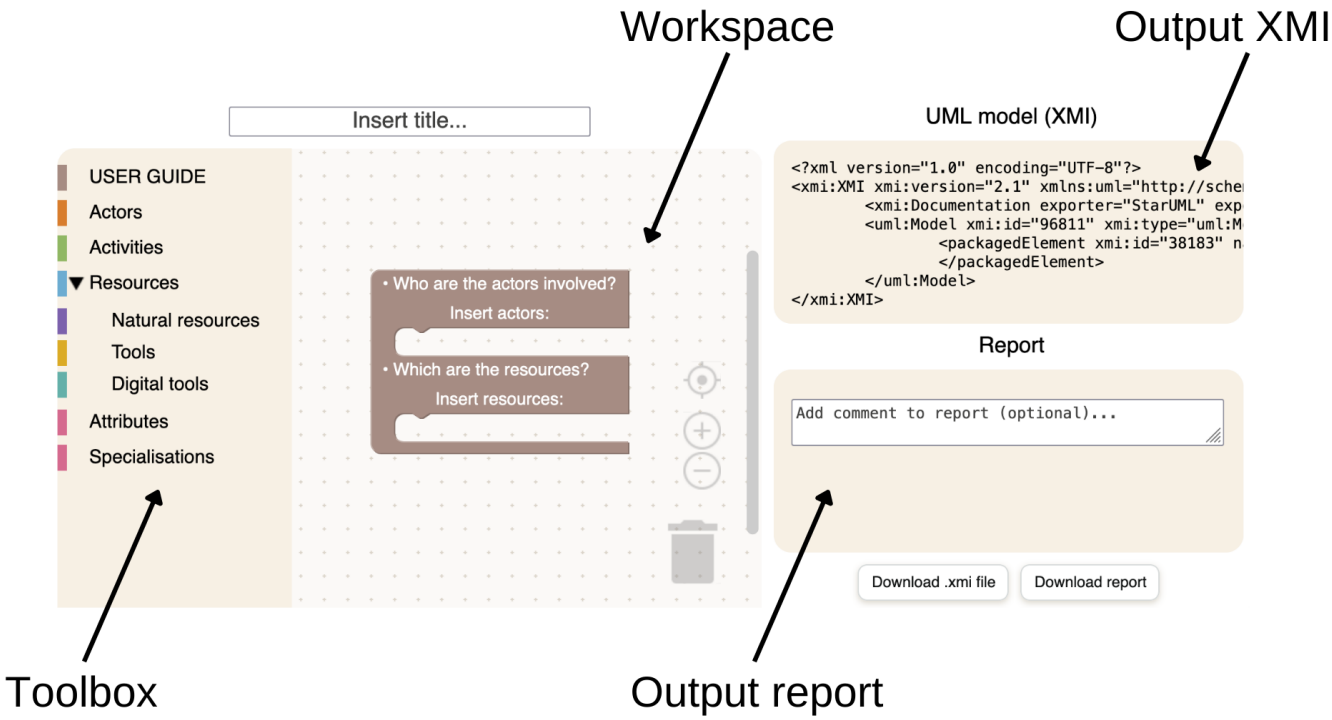

Fig. 1. Structure of the ModeLLer interface with initial set up

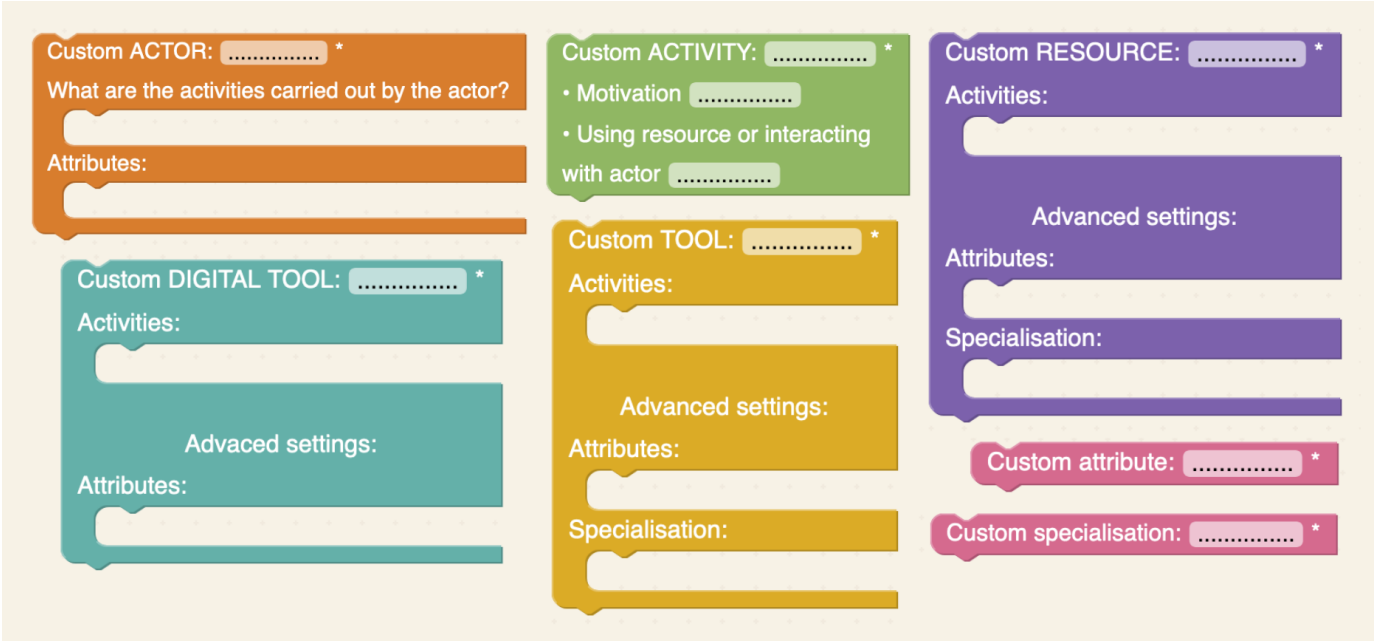

Fig. 2. Additional custom blocks available in ModeLLer

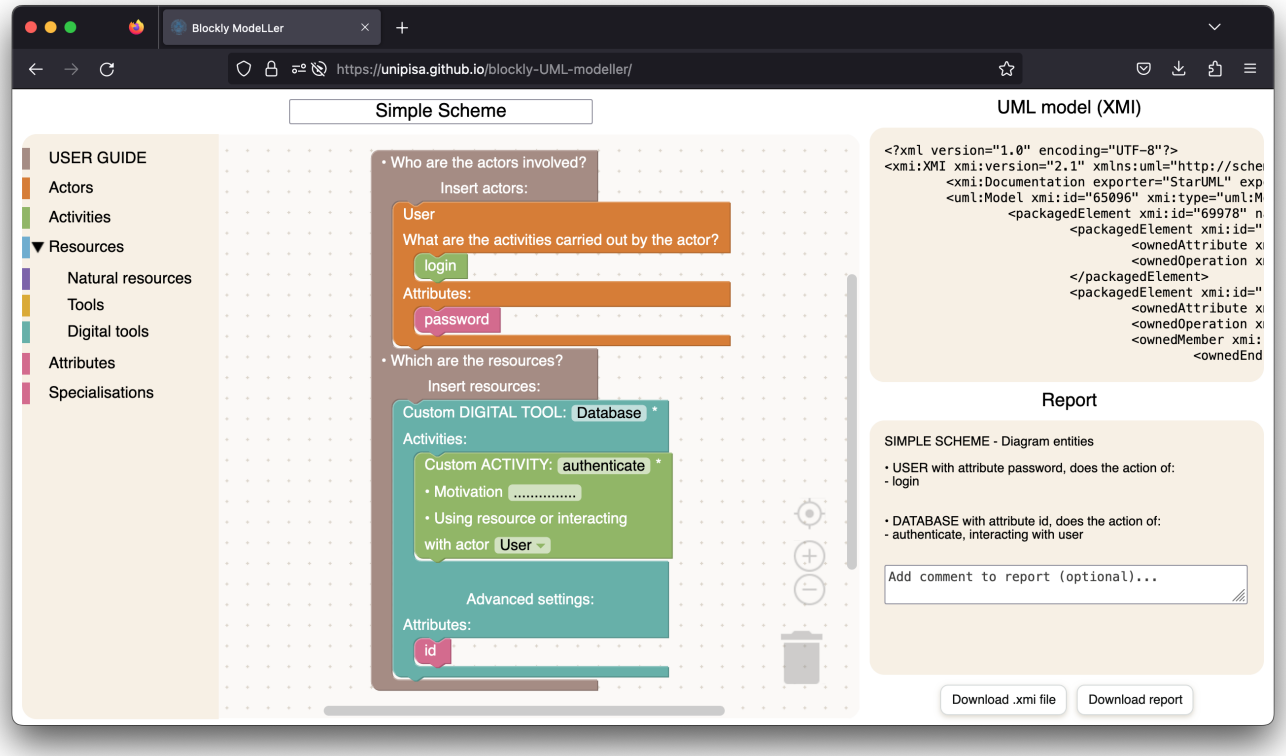

Fig. 3. Example of model

TABLE I
PROTOYPE EVALUATION WITH COGNITIVE WALKTHROUGH

EVALUATION	**Will users try to achieve the right result?**	**Will users notice that the correct action is available?**	**Will users associate the correct action with the result they're trying to achieve?**	**After the action is performed, will users see that progress is made toward the goal?**
ACTION 1: Identifying blocks in toolbox	Yes, from experience	Yes, from experience	Yes, a prompt or label matches action	Yes, there is connection between user goal and system response
ACTION 2: Construction of the model in workspace	Yes, from experience	Yes, they would see a call-to-action	Yes, a prompt or label matches action	Yes, there is connection between user goal and system response
ACTION 3: Creation of advanced structures	No	Yes, they would see a call-to-action	Yes, a prompt or label matches action	Yes, there is connection between user goal and system response
ACTION 4: Export	Yes, the system tells them to	Yes, they would see a call-to-action	Yes, a prompt or label matches action	Yes, there is connection between user goal and system response