

ModeLLer – Enabling End-Users to Model Systems: a Case Study in Digital Agriculture

Chiara Mannari^{1,2}, Elisa Anichini¹, Manlio Bacco², Alessio Ferrari², Tommaso Turchi¹ and Alessio Malizia^{1,3}

Department of Computer Science, University of Pisa, Italy

Institute of Information Science and Technologies "Alessandro Faedo" - ISTI, CNR, Pisa, Italy

Faculty of Logistics, Molde University College, Norway

Abstract

Digital technologies show promising potential in the development of sustainable agriculture. For example, the combination of cloud and edge paradigms, 5G, and the Internet of Things (IoT) allows the development of sophisticated applications – e.g., for food traceability, pest detection, and automatic irrigation – with the possibility to also exploit Artificial Intelligence (AI)-powered techniques. At the same time, digitalisation in agriculture is a socio-technical process that involves several classes of stakeholders with diverse backgrounds and skills, e.g., in farming or technology. Model-driven approaches leveraging diagrammatic notations can support information exchange between different domains. In fact, the development of models can be a co-design practice involving end-users throughout all phases of creation because of their expressive power. However, current modelling platforms are typically oriented toward engineers, and there is a lack of tools accessible to end-users for designing and modelling systems. In this position paper, we present ModeLLer, a prototype of a web environment for modelling systems based on an intuitive visual language that can be exported into standard code. The aim is to increasingly involve users in modelling their digital ecosystems as a task for developing digital applications.

Keywords

end-user development, process modelling, digital agriculture, living labs, meta-design, socio-technical systems

1. Introduction

Digital technologies have a great potential to affect the future of labour. Internet of Things (IoT) components allow the development of sophisticated applications, different techniques based on Artificial Intelligence (AI) contribute to automating manual tasks, and 5G networking will make technology accessible and affordable to users. Tang et al. [1] show the promising potential of digital technologies in the agricultural domain. Furthermore, many solutions are being developed, e.g., for food traceability, pest detection, and smart irrigation aiming to align digitalisation pathways to the *UN 2030 agenda of Sustainable Development Goals (SDGs)* [2]. Recent studies shift the paradigm from a technology-driven to a sustainable one, carrying out interdisciplinary research focusing on the impacts of digitalisation [3] and analysing drivers and barriers [4]. In such contexts, it is increasingly important to include users in participatory activities of design, development and assessment of digital technologies. We carry out our research in the framework of the European project *CODECS – Maximizing the co-benefits of agricultural digitalisation through conducive digital ecosystems* - that aims at developing indicators for sustainable digitalisation starting from the analysis of the transformation occurring to business processes after the introduction of digital technology [5]. To fulfil such objectives in CODECS, an interdisciplinary research group and multiple living labs (LLs), the latter as key research units in the field, have been set up. LLs are networks of farmers, knowledge intermediaries, stakeholders, and policymakers addressing emerging challenges by co-developing user-friendly approaches, methods, and tools in real contexts [6].

Proceedings of the 8th International Workshop on Cultures of Participation in the Digital Age (CoPDA 2024): Differentiating and Deepening the Concept of "End User" in the Digital Age, June 2024, Arenzano, Italy

✉ chiara.mannari@isti.cnr.it (C. Mannari); e.anichini1@studenti.unipi.it (E. Anichini); manlio.bacco@isti.cnr.it (M. Bacco); alessio.ferrari@isti.cnr.it (A. Ferrari); tommaso.turchi@unipi.it (T. Turchi); alessio.malizia@unipi.it (A. Malizia)

🌐 <https://chiaramannari.github.io/> (C. Mannari)

🆔 0000-0002-5488-4150 (C. Mannari); 0000-0001-6733-1873 (M. Bacco); 0000-0002-0636-5663 (A. Ferrari);

0000-0001-6826-9688 (T. Turchi); 0000-0002-2601-7009 (A. Malizia)



© 2024 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

The involvement of multiple stakeholders primarily implies the sharing of knowledge about domains, systems, and processes, and several challenges arise related to information exchange between participants with different backgrounds and expertise. To mitigate issues in communication, we propose the adoption of model-driven requirements engineering techniques [7]. These approaches are used in different stages of the software engineering process and can be particularly effective in exchanging information with users and customers [8]. The most widespread procedure to develop the models is based on the elicitation of knowledge through interviews to domain experts, followed by a phase in which engineers formalise the models through diagrams. In this position paper, we aim to further extend the modelling practice to end-users, i.e., users with no expertise in formal notation by enabling them to generate formal representations of their system. Thus, we address the following research question: *How to make end-users autonomous in the representation of systems without knowing formal languages?* To answer the question, we developed ModeLLer, a prototype of a web tool based on a block-based interface to support modelling by end-users. With ModeLLer, we aim to apply a meta-design approach to LLs. At this stage, we focus on the development of UML class diagrams [9] that can be adopted to represent systems, by providing an overview of the actors, resources, tools, as well as relationships.

2. Methodology

Meta-design and LLs represent the conceptual underpinning of this research. The former, as a framework aimed at defining and creating socio-technical infrastructures in which new forms of collaborative design and development can take place [10]. The latter, as user-centred innovation ecosystems where participants actively collaborate to solve real problems and create new solutions, integrating research and development processes in real contexts [6]. Agricultural LLs are a widespread method for identifying problems and working on common solutions, and new tools are being developed to support participatory activities. An example is provided by the *Living Lab Modeler*¹. The tool offers a digital twin of a LL facilitating the LL assessment, coordination, and dissemination.

As described in a previous work [11], in the framework of CODECS, our research team is working on the development of a method for *socio-technical process modelling* based on a set of different diagrams, namely UML, i* and BPMN to represent the process reengineering after the introduction of digital technology. We plan to apply the method to 20 European LLs with the aim to provide easy-to-read graphical representations of agricultural processes transformed by digital technologies that can be used by different stakeholders to understand how digitalisation may reshape a context. Moreover, the models can be adopted by economists and sociologists for further evaluations such as cost-benefit analysis. Currently, the diagrams are developed through a collaborative procedure involving domain experts and experts on the notations, the former providing the necessary information to create the models and the latter proceeding to the formalisation through specialised software. In a previous paper [12], we described the co-design activity performed in a LL to create a UML class diagram. In this study, we aim to enable domain experts to autonomously create and edit the diagrams. We preliminarily identified a main challenge related to the use of specialised modelling software by end-users; in fact, there is a trade-off between code-oriented software and generalist web platforms, the former is usually complete in representing all language features but has limited user experience². The latter have powerful user experience but usually lack the formal code component³. Furthermore, the design freedom offered by the blank canvas provided by modelling software can pose challenges to users who need to be guided in the creation of structured representations. Thus, the motivation behind the development of ModeLLer is to provide a tool meeting the needs of LLs to model systems and processes without any knowledge of formal notations. To develop the tool, we adopt the SER method for socio-technical systems [13]; the prototype and the walkthrough presented in the next section correspond to the first seeding phase and future work will include the evolution of the prototype following users' priorities.

¹<https://www.livinglabmodeler.eu>

²See for example Eclipse Papyrus <https://www.eclipse.org/papyrus/>, accessed 29 March 2024

³See for example Lucidspark <https://lucidspark.com/>, accessed 29 March 2024

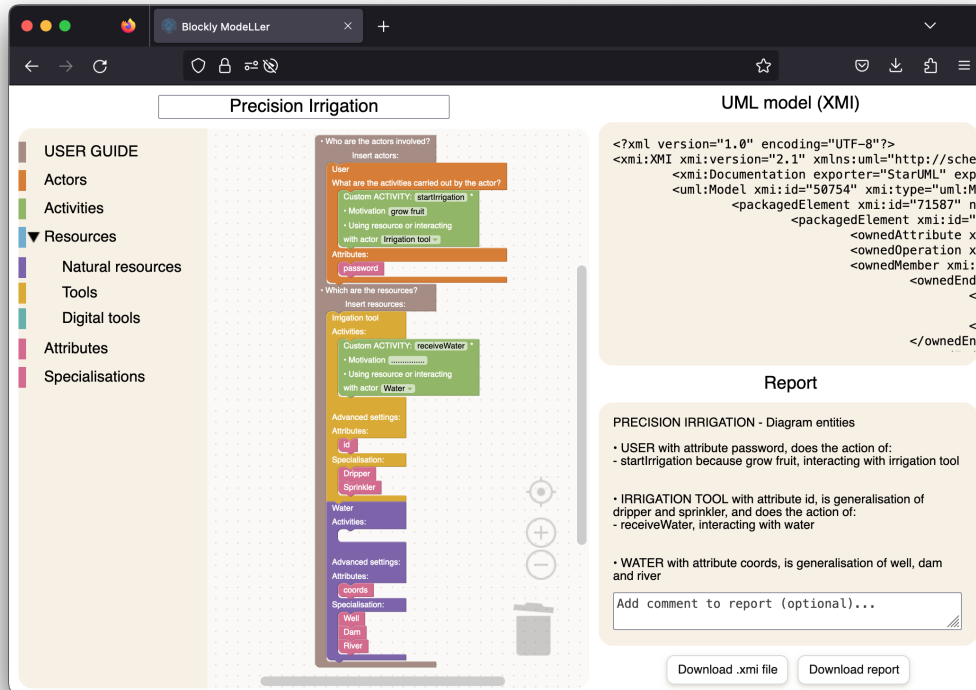


Figure 1: Creation of precision irrigation model with ModeLLer

3. The ModeLLer Prototype

ModeLLer is a prototype of a web environment for modelling systems based on a block-based visual language which can be exported into standard code⁴.

Currently, the functionalities offered by the prototype allow the creation of UML class diagrams through the interaction with a block-based visual editor (Fig. 1) in three linear steps:

1. **Explore.** The user can explore available blocks contained in the toolbox panel and divided into various categories.
2. **Drag-and-drop.** It is possible to drag and drop blocks to the workspace following a guided procedure and create a structure consisting of several interlocking blocks.
3. **Export.** At any time, the user can download a file in XMI format containing the code to be imported into a UML design software and a textual report describing the structure created.

As previously mentioned, according to our method, multiple diagrams are expected to reach a complete representation of a process. However, in developing the prototype, we chose the UML class diagram as a starting point. Offering a preliminary overview of the process, the class diagram focuses on the process structure, i.e., the system, by representing all the entities involved: actors, resources, technological components, as well as operations and relations among entities.

The tool we selected for developing the block-based visual language in ModeLLer is Blockly [14]. Blockly is a JavaScript library based on a no-code/low-code approach [15] that is commonly adopted for simple and user-friendly development environments [16]. Recent studies have proposed block-based solutions to assist users in customising applications for monitoring and configuring agricultural processes. [17, 18] Although Blockly provides pre-defined blocks, these are typically programming-oriented, and additional custom blocks were necessary to enable the creation of the class diagram and

⁴ModeLLer code, as well as a demo and a video presentation are available on GitHub: <https://github.com/Unipisa/blockly-UML-modeller>

support users in identifying all the necessary elements. In ModeLLer, blocks represent the different entities involved in the system and differ in colour, size and number of interlocking blocks. For example, the orange custom block *Actor* represents the actor class and contains a textual attribute to be assigned by the user. This text can be the actor's role (e.g., farmer, employee, drone pilot, etc.). Actors must be interlocked with one or more *Activity* blocks as suggested by the label placed on the element. The creation of a diagram will be illustrated in detail in the next section.

4. A Case Study on Precision Irrigation

We selected the UML class diagram of a precision irrigation system as a case study for testing the ModeLLer prototype. The diagram is extracted from the pilot study developed for setting up the modelling method presented in [11]. The diagram was developed by experts on the notation based on a report collaboratively written by the interdisciplinary team involved in the LL and includes all the entities to be considered in representing the irrigation process, i.e., a user, a decision support system, natural resources and irrigation tools.

Interacting with the current version of ModeLLer, it is possible to rebuild the precision irrigation class diagram in two consecutive steps: (1) creation of the classes, attributes, operations and relationships with the block-based editor and export of XMI code; (2) import of XMI code in a UML application and visualisation of the diagram.

The creation of a model starts from the questions contained on the preset blocks appearing on the initial setup of the workspace: (1) *Who are the actors involved in the process?* and (2) *Which are the resources?*. The user can select available blocks in the toolbox and drag and drop them to the workspace. Once an actor block is added to the workspace, the user is asked a new question: (3) *What are the activities carried out by the actor?* The user is asked to provide information on activities performed following this pattern: an *<actor>* performs an *<activity>* using *<resource>* to *<motivation>*.

Advanced UML structures, such as composition, generalisation, and attributes, are also available in the toolbox or directly on the blocks. Once the modelling is finished, as represented in Fig. 1, it is possible to download the XMI file containing the code generated while the model is created on the workspace. Along with the XMI code, a textual report is generated and available for download. The textual report serves as support explaining the structure created by the user through natural language. To obtain a diagram in the UML graphical notation, it is necessary to perform step 2, importing the XMI file into a UML design software supporting XMI standard⁵. A syntactically correct class diagram can be created by selecting the XMI imported classes, which maintain features and relationships as defined by the user in ModeLLer. In future releases, we will work to include, in addition to XMI code, the graphical notation and export the image format directly from ModeLLer.

5. Evaluation and Re-design

The prototype was evaluated through the cognitive walkthrough approach [19]. This technique allowed us to evaluate the learnability and usability of ModeLLer from the user's point of view.

A remote workshop was conducted with a team of cross-functional evaluators to assess various aspects of the prototype interface. This team included three males and one female expert, aged between 35 and 45, with backgrounds in formal languages, visual languages, web design, and agricultural technology. They were guided through a series of tasks related to the modelling activity by a facilitator, answering predefined questions. The findings of this assessment are detailed in Table 1, which lists the actions reviewed and the experts' responses.

The first action analysed by the group of experts was the *identification of blocks* within the toolbox. Overall, the experts agreed that "Finding blocks is simple and intuitive. The menu, divided into different categories, guides the user towards identifying the blocks". By clicking on the label of the chosen

⁵We tested XMI import in StarUML v.5.1.0 <https://staruml.io/> and Visual Paradigm 17.0 <https://www.visual-paradigm.com/>, last accessed 29 March 2024

Table 1
 Prototype 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

category, the system responds by opening a panel containing the blocks of the selected category. In this way, the user receives clear feedback to have performed the correct action.

The second action analysed by the experts was the *graphic construction of the model*, i.e., dragging the blocks into the workspace and fitting them together. Even for this action, the review team expressed itself positively. The presence of a fixed block in the work area containing the phrases *Insert actors* and *Insert resources* with space underneath for inserting something suggests the user to insert other blocks. The fact that the words *Actors* and *Resources* are also the names of the two main categories guides the user to open these categories in the toolbox. Once a category is opened, hovering over a block will cause the block border to glow yellow and the cursor to a hand ready to select, move and release. Thus, the group agreed, “It is obvious to the user that the block editor has a drag-and-drop mechanism, and the interlocking on the blocks, similar to that of the puzzle pieces, acts as a sort of visual call-to-action”.

The third activity that was analysed pertains to the *creation of advanced structures*. This may correspond to a generalization, an aggregation, or an association between two classes; the latter case was examined during the workshop. In the prototype, as well as in traditional UML design software, to obtain an association between two classes, both class blocks must be connected. The perplexities of the experts concerned the insertion of the class name in a text field appearing under the label *Using resource or interacting with actor* of the Custom activity block. The team agreed that “This action requires too much cognitive effort for the user to achieve the right result”. The group thought about alternative solutions.

The last examined action was the *export of the XMI file* and textual report. For the action analysed in this step, the group was unanimously positive: “The two download buttons are standard and easily visible, and give the idea of being clickable, as their labels make it immediately clear what action will be performed”. When one of the two buttons is clicked, the system responds by starting the download of the chosen file and opening the start download window in the browser; then the progress bar allows the user to understand that the action was performed correctly and is achieving its goal.

The feedback collected during the cognitive walkthrough workshop helped us to re-design critical emerging components. The main change was related to the third action, *creation of advanced structures*, which was the only activity evaluated negatively by the experts. The solution identified consisted of replacing the text field associated with the input *Using resource or interacting with actor* with a selector. During the creation of an association between two classes, it is now possible for the user to choose the class to be associated from a list containing the already existing classes instead of manually typing the name of the desired class. This suggests possible relations without seeking the element name in the diagram while limiting the user’s errors.

6. Limitations and Future Work

This research acknowledges several limitations, which provide a roadmap for future work.

Firstly, a significant limitation lies in the current stage of development. ModelLer, which is still in its initial development phase, offers limited functionalities at this time. Primarily, the tool is confined to creating entity models, i.e., UML class diagrams, with additional aspects of the system, like process modelling, requiring further enhancement. As the tool is part of a European research project, we anticipate incorporating more features and improvements in subsequent versions.

Secondly, the preliminary evaluation was conducted with a cross-functional team of experts from fields such as visual languages, formal languages, and agricultural technology. However, these evaluators may not reflect the intended end-user demographic of the tool. Furthermore, the evaluation was restricted in scope because of the tool's early development stage, thus lacking the depth and breadth that a comprehensive usability study might provide, including individualised feedback. Therefore, future studies will involve key stakeholders in the evaluation stages to gain a more holistic understanding of the tool's usability and areas for improvement.

Thirdly, ModelLer has been designed with a specific focus on digital agriculture, meaning that its broader applicability across other domains has not been explored. Future research could investigate this tool's potential utility across various fields beyond digital agriculture.

Lastly, the integration and application of the concepts of Meta-Design and Living Labs in our work is still at a preliminary stage. Further work is needed to deepen the integration of these concepts in our model-driven requirements engineering techniques and to better clarify their relationship with the broader research environment.

Moreover, in future work, it is worth exploring the potential of the tool for synergistic human-AI co-creation through natural language interactions and generative AI.

Acknowledgment

Research partly funded by the European Union's Horizon Europe research and innovation programme under grant agreement no. 101060179.

References

- [1] Y. Tang, S. Dananjayan, C. Hou, Q. Guo, S. Luo, Y. He, A survey on the 5g network and its impact on agriculture: Challenges and opportunities, CEA 180 (2021).
- [2] R. Bertoglio, C. Corbo, F. M. Renga, M. Matteucci, The digital agricultural revolution: A bibliometric analysis literature review, IEEE Access 9 (2021).
- [3] K. Rijswijk, L. Klerkx, M. Bacco, F. Bartolini, E. Bulten, L. Debruyne, J. Dessein, I. Scotti, G. Brunori, Digital Transformation of Agriculture and Rural Areas: a Socio-Cyber-Physical System Framework to Support Responsibilisation, JRS (2021).
- [4] A. Ferrari, M. Bacco, K. Gaber, A. Jedlitschka, S. Hess, J. Kaipainen, P. Koltsida, E. Toli, G. Brunori, Drivers, Barriers and Impacts of Digitalisation in Rural Areas from the Viewpoint of Experts, IST 145 (2022) 106816.
- [5] CODECS, <https://www.horizoncodecs.eu/>, 2023.
- [6] S. Leminen, M. Westerlund, A.-G. Nyström, Living labs as open-innovation networks (2012).
- [7] G. Loniewski, E. Insfran, S. Abrahão, A systematic review of the use of requirements engineering techniques in model-driven development, in: MODELS 2010, Springer, 2010, pp. 213–227.
- [8] D. Moody, The “physics” of notations: Toward a scientific basis for constructing visual notations in software engineering, IEEE Transactions on Software Engineering 35 (2009) 756–779.
- [9] Unified Modeling Language (UML) 2.5.1 Core Specification, <https://www.omg.org/spec/UML>, 2017.
- [10] G. Fischer, E. Giaccardi, Meta-design: A framework for the future of end-user development, End user development (2006) 427–457.

- [11] C. Mannari, G. O. Spagnolo, M. Bacco, A. Malizia, Digitalisation of agriculture: Development and evaluation of a model-based requirements engineering process, in: REFSQ, 2023.
- [12] C. Mannari, M. Bacco, A. Ferrari, L. Ortolani, M. B. Lai, C. Mignani, A. Silvi, A. Malizia, G. Brunori, A methodology for process modelling in living labs to foster agricultural digitalisation, in: 2023 IEEE MetroAgriFor, 2023.
- [13] G. Fischer, T. Herrmann, Socio-technical systems: A meta-design perspective, IJSKD 3 (2011) 1–33.
- [14] Blockly, <https://developers.google.com/blockly?hl=it>, 2022.
- [15] A. C. Bock, U. Frank, Low-code platform, Business & Information Systems Engineering 63 (2021) 733–740.
- [16] M. Resnick, J. Maloney, A. Monroy-Hernández, N. Rusk, E. Eastmond, K. Brennan, A. Millner, E. Rosenbaum, J. Silver, B. Silverman, et al., Scratch: programming for all, Communications of the ACM 52 (2009) 60–67.
- [17] B. M. A. Amer, H. Chouikhi, Smartphone application using a visual programming language to compute drying/solar drying characteristics of agricultural products, Sustainability 12 (2020).
- [18] G. Abraham, R. R., M. Nithya, Smart agriculture based on iot and machine learning, 2021, pp. 414–419. doi:10.1109/ICCMC51019.2021.9418392.
- [19] Nielsen Norman Group, “Evaluate interface learnability with cognitive walkthroughs”, <https://www.nngroup.com/articles/cognitive-walkthroughs>, 2022.