

End-user Requirements Modelling: an Experience Report from Digital Agriculture

Chiara Mannari^{1,2}, Mino Sportelli¹, Harika Meesala³, Ogochukwu Felicitas Okoye³, Fabio Lepore³, Manlio Bacco¹, Gianluca Brunori³, Alessio Malizia^{2,4},
and Alessio Ferrari¹

¹ Institute of Information Science and Technologies “A. Faedo”, CNR, Pisa, Italy

² Computer Science Department, University of Pisa, Pisa, Italy

³ Department of Agriculture, Food and Environment, University of Pisa, Pisa, Italy

⁴ Faculty of Logistics, Molde University College, Molde, Norway

Abstract. Context and motivation: End-user development focuses on enabling non-professional programmers to create or extend software applications on their own. However, before beginning the development process, software engineering best practices recommend performing requirements engineering (RE) activities, including requirements modelling. **Question/problem:** There is limited research on how end-users can model system requirements. **Principal ideas/results:** In this experience report, we investigate the problem of end-user requirements modelling in an EU-funded project about agricultural digitalisation. Specifically, a team of agronomists was directly involved in the creation of UML, iStar, and BPMN diagrams to model the transformation of socio-technical processes in four different concrete scenarios. They followed a formalisation procedure proposed within an RE method designed to help stakeholders evaluate the impact of agricultural digitalisation. Starting from textual reports including a description of the process as-is and the process-to-be, they followed step-by-step guidelines for model creation. **Contribution:** This paper reports insights from the experience from the viewpoint of the agronomists and software engineers involved. We identify eight key lessons that highlight the added value of end-user requirements modelling for achieving a shared and in-depth understanding of the socio-technical processes under analysis.

1 Introduction

Digitalisation is a socio-technical process that drives significant transformation, with impacts that can be assessed from various perspectives, including social, institutional, economic, environmental, and technological [9, 10]. This is especially relevant in agriculture, where traditional practices are undergoing a disruptive shift due to digital technology adoption, e.g., automatic irrigation, internet of things (IoT) for soil observation, unmanned aerial vehicles (UAVs) for crop monitoring and spraying [21]. Recent research stresses the importance of interdisciplinary approaches in developing innovative agricultural solutions [22, 7] and emphasises the need to involve end-users in the development of the technology [6].

End-user development (EUD) [16] entails empowering non-professional developers to create or modify software artefacts to meet their own specific needs. However, before development, software engineering best practices recommend performing requirements engineering (RE) activities, including requirements modelling. Model-driven requirements engineering (MoDRE) techniques [14], such as iStar [25], UML [3], and BPMN [2], use diagrammatic notations to capture different aspects of system requirements, such as functionality, structure, goals, data, processes, and workflows [5]. We argue that these techniques can be used to support end-user requirements modelling, i.e., involving non-professional requirements engineers in representing systems and processes in their specific domain.

In this paper, we share the experience of a team of agronomists who applied a formalisation procedure to create graphical models with MoDRE notations, namely iStar, UML, and BPMN, with the support of guidelines produced by software engineers. The models aimed at representing the process transformation due to digitalisation in four different real-world cases. The procedure is part of a RE method developed in a design science study [23] in the context of a Horizon Europe (HE) CODECS project (Maximizing the co-benefits of agricultural digitalisation through conducive digital ecosystems) [1]. In CODECS, we interact with 20 Living Labs (LLs) [15], i.e., communities of local practices, including farmers, knowledge intermediaries, stakeholders, and policymakers carrying out co-design activities for addressing common goals. One of the objectives of CODECS is to use general-purpose MoDRE techniques to enable stakeholders to reason about the costs and benefits of digitalisation in agriculture. From the end-user modelling experience, we derive a set of lessons learnt from the viewpoint of the agronomists and the software engineers involved. The agronomists were able to use the notations by following the provided guidelines, with minimal interaction with the software engineers and only a few syntactic errors. They also expressed the intention to reuse the approach in the future. The software engineers agree that they would not have been able to produce models with the same degree of accuracy on their own since they lack the specific domain knowledge required to fully understand the context. On the other hand, the agronomists remarked on the substantial abstraction effort required and complained about the limitations of the tools, which suggest the need for more user-friendly solutions. This paper contributes to the literature with one of the first experiences in end-user modelling through the concrete application of MoDRE techniques, addressing the gap of limited real-world studies in this field [20]. Supplementary material is provided in [18].

2 Modelling Approach and Guidelines

The CODECS project engages stakeholders in evaluating the costs and benefits of agricultural digitalisation through interactions with 20 living labs (LLs) across Europe. A key task involves developing graphical models to represent process transformations, facilitating the analysis of socio-economic and environmental impacts of introducing new technologies. These models are expected to provide a stakeholder-friendly alternative to detailed, time-consuming methods like lifecycle assessment [13].

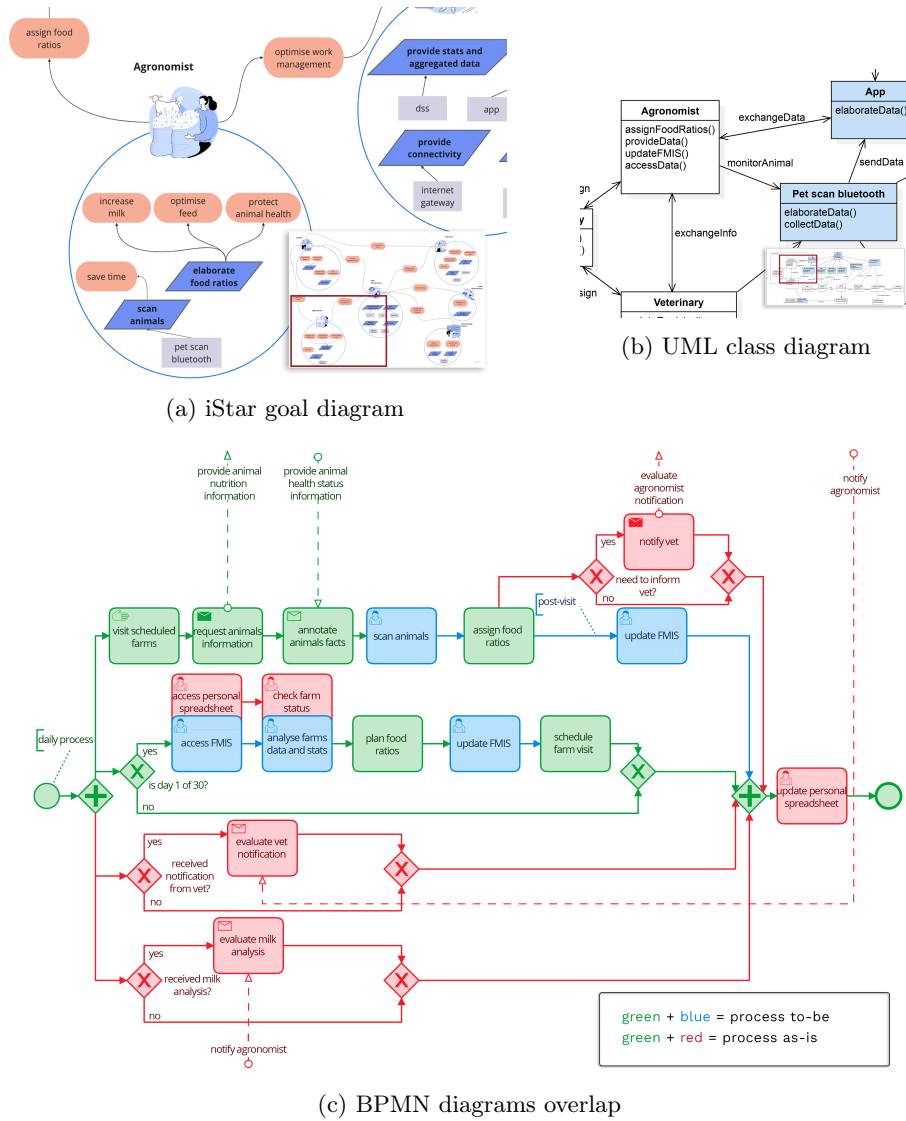


Fig. 1: Models representing part of a milk production process transformed by the introduction of a farm management information system.

To develop the graphical models, we are performing a design science study [23], in which we have defined a method for collecting process-related data from LLs, and for representing the process transformation using MoDRE notations [17]. The adopted MoDRE notations are UML class diagrams, iStar, and BPMN. Fig. 1 shows an excerpt of the diagrams based on a pilot study developed in a CODECS LL, in which a farm management information system is introduced to monitor milk production in a cheese-making process. These diagrams were

Table 1: Summary of formalisation procedure for iStar and UML models

Goal Diagram - iStar

- Actors** - Select actors from [ACTORS PROCESS AFTER] and create an Actor boundary.
 - Select technology from [DIGITAL SOLUTION] and create an Actor boundary (Technology boundary); possibly place it at the centre of the diagram.
- Goals** - For each actor select goal list from [RATIONALE] and [ADVANTAGES].
 - Classify and create goals. Place goals in internal, actor-actor or actor-technology positions. Express goals through a short sentence containing a verb.
- Tasks** - For each actor select main activities relevant to strategic aspects from [ACTIVITIES PROCESS AFTER] and abstract tasks. Place tasks inside the actor's boundaries.
- Tools** - Select technological resources from [RESOURCES] and [SOLUTION]; create and place tools inside the actor's boundaries, below the correspondent task.
- Connections** - *Tool-task*: for each tool, add a link between the tool and its corresponding task with the arrow directed from the tool to the task.
 - *Task-goal*: for each task add a link between the task and its corresponding goal with the arrow directed from the task to the goal.
 - *Actor-actor goal*: for each actor-actor goal, add a link connecting the two actor boundaries and the goal. The arrow should be directed from the active actor, e.g., the actor performing actions to fulfil the goal, to the recipient actor. If the actors share the same goal, put the arrow in both directions.
 - *Actor-technology goal*: for each actor-technology goal, add a link connecting the actor boundary to the technology boundary with the arrow directed from the actor to the technology.

Structure Diagram - UML

- Classes** - Select the technological system from [SOLUTION] and represent it as a class.
 - Identify components (hardware and software) of the tech. system from [SOLUTION] and [RESOURCES]; represent them as class and link with aggregation, (i.e., diamond).
 - Select actors from [ACTORS] and resources from [RESOURCES] and represent them as classes.
- Connections** - *Actor-technology/Actor-resource*: Add a link between the classes based on [PROCESS AFTER]. Add a label on the link with the action performed by a class actor using a technology or resource class. The arrow direction should go from the actor to the technology or resource.
 - *Actor-actor*: if actors have direct interactions, add a link between the actors based on [PROCESS AFTER]. The arrow direction should go from the active actor to the passive.

developed by the 1st author, a software engineer. In Fig. 1a, the iStar diagram represents the goals, activities and relationships of the agronomist in the cheese-making process. Fig. 1b contains a portion of the UML class diagram with a detailed view of the class Agronomist and neighbouring elements. The new classes introduced by digital technology are in light blue. Fig. 1c is an overlap of two BPMN models focused on the pool of the agronomist and showing the process transformation for this actor. Details on the diagrams, the pilot case, and the structure of the design study are reported in a previous paper [17].

The information to produce the models has been collected by LL coordinators following a set of guidelines and a reporting template developed in the context of the project. The template includes different sections, indicating the technological solution adopted by the LL, its advantages, the actors involved, and other process-relevant information. This part of the study is included in a public CODECS deliverable that will be published in the near future. In the current study, we present the experience of applying a structured procedure to enable end-users to perform the modelling activity. The main guidelines of this procedure, illustrated in a 22-page document shared in the supplementary material, are summarised in Table 1 and 2. The document is practice-oriented and includes guidelines to create model elements (goals, tasks, etc.) and connect them, based on the content of the reporting template used for data collection. To this end, the document includes specific references to the template. In Table 1 and 2, we simplify these references with the placeholder [TITLE] indicating the

Table 2: Summary of formalisation procedure for BPMN models

Diagram 1: BPMN process after	
Actors	- Select actors from [PROCESS AFTER] and create a pool for each actor. Select technology in [SOLUTION] and create a pool.
Start/End process	- Add a start event and an end event inside each pool. Add a text annotation on the start event and indicate the execution frequency.
Tasks	- Select the activities in [PROCESS AFTER] and create a sequence of connected tasks inside each actor boundary and within the start and end event. Optionally, assign a type to each task choosing among: manual, system, receive, send.
Connections	- Identify send tasks and create a link starting from the send task to the corresponding recipient's receive task / pool. In case the send task triggers the beginning of the recipient's process, transform the start event of the recipient into a message start event and connect the send task. - Add a label on the dotted line with the object of the activity.
Gateways	- <i>Parallel gateways</i> : from [PROCESS AFTER] extract activities that are executed in parallel and place a parallel gateway before the first task executed to create multiple paths. Add a task or a sequence of tasks to each path. Repeat the gateway after the last tasks. - <i>Exclusive gateways</i> : from [PROCESS AFTER] extract activities that are executed optionally and place an exclusive gateway to create multiple paths. Add a task or a sequence of tasks to each path. Add a label on the gateway explaining the condition and add a label on each path explaining the verification of the condition. Repeat the gateway after the last tasks.
Annotations	- Add annotations on tasks to express additional information in [PROCESS AFTER].
Diagram 2 - BPMN process before	
Actors	- Copy in a new file the model(s) created in Diagram 1: BPMN process after - Delete the pools of actors that were not in [PROCESS BEFORE]. - Delete or edit the technology pool according to the info in [SOLUTION]. - Select actors from [PROCESS BEFORE] that are not in [PROCESS AFTER] and create new pools. Add start and end events inside the pool.
Tasks, connections, gateways	- Select activities from [PROCESS BEFORE] and edit the pool of each actor with corresponding tasks, relationships, gateways and annotations following the instructions provided in Diagram 1: BPMN process after.
Diagram 3: BPMN process overlap	
	- Copy in a new file the model(s) created in Diagram1: BPMN process before. Paste the content of Diagram2: BPMN process after. Place Diagram2 next to Diagram1 and compare the two processes. - Colour in green all the elements of Diagram1 that do not change in Diagram2. - Colour in red all the elements of Diagram1 that are no longer present in Diagram2. - Copy-paste all the new elements from Diagram2, colour in blue and overlap to Diagram1. - Remove Diagram2.

title of a section of the data collection report, e.g., [ACTORS PROCESS AFTER], [SOLUTION]. The modelling guideline document also includes examples, as well as an introduction to the modelling languages and recommended tools (StarUML for UML class diagrams, Camunda for BPMN, and LucidSpark for iStar). Additionally, the document provides links to template files to produce the models with the aforementioned tools.

3 Application of the End-user Modelling Guidelines

To apply the proposed approach, we set up an interdisciplinary team, with half the members being software engineers—1st, 6th, and 9th author—and the other half being agronomists—2nd to 5th author. The agronomists are researchers with 3 to 4 years of experience in the field, with a specific focus on agricultural economics and agritech. The software engineers have 7 to 18 years of research and practical experience in RE and end-user design. The software engineers developed the method through multiple iterations. They first developed models for two pilot studies, also involving other agronomists to get feedback on the models (cf. [19, 17]). Then, they performed internal focus groups to “reverse-engineer” the procedure followed to create guidelines that were sequential, structured, and

suitable for end-users. Each agronomist, who had no previous experience in the notations, applied the procedure to one of the four cases described below.

3.1 Selected Cases

Each agronomist was assigned a case from one of the CODECS LLs. The cases are concerned with technologies such as UAV, IoT, image processing, and farm management information systems. They were selected according to the agronomists' backgrounds and research interests and to ensure case and EU region diversity. In the following, we provide a summary of each case.

Case 1 - Unmanned Aerial Vehicles (UAVs) for grape harvesting. Case 1 examines an agroecological vineyard located in Italy. This 10-hectare farm, with 3.3 hectares dedicated to organic and biodynamic-certified vineyards, operates under strict fertiliser and pesticide limits. Climate change has reduced both the yield and wine quality, challenging the farm's competitiveness. In response, it has adopted digital technologies, including UAVs and sensor data, to optimize grape harvesting based on quality. UAVs enable staged harvesting by processing higher-quality grapes first. New actors, such as technicians operating the UAVs and generating fertility and vigour maps, now play a role in the grape harvesting process. Maps, accessible via mobile software, assist the farmer in coordinating the harvest and improving grape selection, ultimately enhancing wine quality.

Case 2 - Ultra-High-Resolution RGB imagery for early detection of potential outbreaks Case 2 concerns a LL in Belgium, which develops digital solutions to reduce chemical plant protection products in arable farming. To achieve this, the LL promotes targeted chemical applications using ultra-high-resolution RGB images for early detection of pests, diseases, and weeds. Currently, it is testing a prototype of the technology, which utilises multiple sensors, UAVs for image collection, robust AI models and data processing software for accurate detection and mapping of potential outbreaks. A task map is then generated for site-specific spraying.

Case 3: Remote agricultural monitoring and advisory system (RAMAS) Case 3 is concerned with RAMAS, a Farm Management Information System (FMIS) developed within a LL in North Macedonia to support efficient and data-driven farming. Acting as a centralised hub and data repository, the system integrates advanced technologies—including IoT devices, UAVs (drones), satellite imagery, and soil analysis reports—to support the cultivation of barley, corn, hazelnuts, and vineyards. The platform can be accessed by farmers and advisors, improving communication, enhancing resource efficiency, and addressing the limitations of manual record-keeping.

Case 4: Electronic milk meters and DSS for sheep breeding Case 4 looks at the digitalisation of livestock farming in sheep breeding for the production of Roquefort PDO cheese in a French LL. The proposed solution integrates multiple technologies: a) electronic milk meters to monitor the amount produced by each animal using flow sensors; b) a self-moving automatic feed distribution system; c) sensors for monitoring humidity and temperature. The goal of the solution is supporting farmers in optimising the feeding according to their productivity and physiological attributes, serving as decision support systems.

Case	Data collection	Models	Revision
1	Report: #14pp Stakeholders: #28 (res., farm., tech., cus.) Activities: I (7), VIS (7)	iStar: #44 el. UML: #19 el. BPMN #72 el.	iStar: Err. #1; Warn. #3 UML: Err. #0; Warn. #4 BPMN: Err. #1; Warn. #2
2	Report: 18pp Stakeholders: #12 (farm., tech., pol.) Activities: I (6), FG (1), VIS (4), WS (1)	iStar: #123 el. UML: #63 el. BPMN #402 el.	iStar: Err. #3; Warn. #7 UML: Err. #1; Warn. #1 BPMN: Err. #2; Warn. #4
3	Report: 22pp Stakeholders: #12 (res., adv., farm., pol.) Activities: I (1), FG (4), VIS (1), WS (1)	iStar: #43 el. UML: #24 el. BPMN #50 el.	iStar: Err. #1; Warn. #6 UML: Err. #1; Warn. #4 BPMN: Err. #1; Warn. #3
4	Report: 19pp Stakeholders: #31 (res., adv., farm., inst.) Activities: I (4), FG (2), VIS (1)	iStar: #86 el. UML: #30 el. BPMN #130 el.	iStar: Err. #1; Warn. #7 UML: Err. #1; Warn. #4 BPMN: Err. #1; Warn. #2

Table 3: Summary of data collection, modelling and revision. Abbreviations: I = interview, FG = focus group, VIS = visit, WS = workshop, res. = researcher, adv. = advisor, farm. = farmer, inst. = institution, prod. = producer, tech. = technology provider, pol. = policymaker, cus. = customer

3.2 Data Collection and Guidelines Application

According to the CODECS timeline, the collection of the data to be used for modelling was carried out by the LL coordinators from October 2022 to May 2024. LL coordinators interacted with multiple stakeholders through different activities, such as interviews, focus groups, and workshops. Then, they filled in the reporting templates and delivered the reports in May 2024. The software engineers organised a two-hour remote meeting on 14th June 2024 to present the modelling guidelines to the agronomists and agree on a timeline for the modelling activity. The agronomists used the guidelines to produce the models based on the LL reports. The models were then revised by the software engineers for syntactic errors—the models will be validated for completeness with the LL coordinators in future project activities. Discussion meetings on the 5th July and 11th October 2024 were carried out involving all the participants, who discussed the models and exchanged viewpoints on the experience. Joint writing sessions were then organised to summarise the lessons learnt.

Table 3 outlines the results of data collection, modelling, and revision activities. The data collection involved about 20 subjects on average per LL, with diverse profiles (farmers, advisors, researchers, technology providers, and institutions). The models were quite detailed and extensive, with an average of 74 elements for iStar, 34 for UML, and 164 for BPMN. The revision distinguished between errors (i.e., formal syntactic and semantic mistakes diverging from the standard of the notations, e.g., wrong type of symbols) and warnings (i.e., less severe issues associated with stylistic reasons, e.g., overlapping lines). Table 3, col. Revision, reports unique instances of these issues—each type of systematic error is counted once. We notice that these are limited, with only 1 to 3 types of errors per diagram.

The lessons learnt are presented in the following section. The models are shared in our external repository, together with the revision documents.

4 Lessons Learnt

The lessons learnt reflect the viewpoints of the software engineers and agronomists involved in the activity. Since these individuals are also the authors of the paper,

and given that this is an experience report, we do not provide specific quotations from the discussions. Instead, we present the team’s synthesised and collectively agreed-upon reflections.

L1 – Practice-oriented guidelines enable end-users to model complex requirements The agronomists received only an initial illustration of the modelling guidelines, and then they were given the guideline document. The agronomists confirmed that they were able to follow the instructions provided without encountering major obstacles. They remarked on the effectiveness of the guidelines, especially for their practice-oriented structure, and the presence of examples. The document is seen as beneficial for understanding the objective of the activity, the principles of the languages, and the specific steps for systematically creating the diagrams⁵. The success of the experience is highlighted by the complexity of the models produced in terms of the number of elements, and by the limited number of errors. On the other hand, the agronomists agree that a preliminary hands-on session under the guidance of the software engineers could have helped them reduce the learning effort.

💡 This lesson underscores end-user ability in requirements modelling and formal accuracy even in complex socio-technical contexts like agricultural LLS. It highlights that the practice-oriented guidelines proposed, which are arguably simple and do not require a deep knowledge of the languages, can lead to successful modelling.

L2 – End-users have different learning approaches The agronomists shared their strategies for addressing the modelling task. While they used the modelling guidelines as a primary reference, they adopted different approaches according to their learning preferences and attitudes: some relied solely on the documentation following a learning-by-doing method and waiting for the revision by the software engineers; others supplemented the guidelines with additional explanations of the notations, such as the video tutorials on BPMN offered by the Camunda platform, also to understand more theoretical principles behind the modelling languages. This is in line with observations from Felder [8], who emphasises that aligning teaching methods (e.g., methodology-oriented, practice-oriented) with learning styles (e.g., active vs reflective, sequential vs global) can improve learners’ engagement and comprehension.

💡 By supporting diverse learning methods, we can help end-users effectively perform the modelling task. This highlights the need for adaptable training resources that complement the modelling guidelines.

L3 – Abstraction is regarded as the most challenging task The agronomists generally found the modelling activity to be time-consuming, requiring an

⁵ This opinion may be affected by the Hawthorne effect, as the authors of the paper are also those who developed the guidelines, and the agronomists may not feel comfortable in expressing negative judgments. However, they are well aware that this activity is oriented to validate the approach and that negative opinions can be beneficial to improve it.

average of 12 hours, spread across multiple sessions with breaks. Although the guidelines allowed them to speed up the practical tasks, they declared that the most challenging part of modelling is related to interpretation and abstraction, consisting of making clear-cut choices about relevant information and finding the right words to name elements in the diagrams. They agree that this activity requires mental effort and flexibility, necessitating constant revision to achieve coherence in the representation. Abstraction is a key computational thinking skill [24]; it often requires specific training and experience, which may not traditionally be part of an agronomist’s background.

💡 While practical aspects of model creation can be supported with guidelines and tools, the task of abstracting information is the most challenging demanding cognitive effort and computational thinking skills.

L4 – Domain expertise can fill the gaps of incomplete information elicitation. Participants highlighted that their background in agronomy helped them understand the reports from the LLs and fill in the gaps where information was missing or inconsistent. Challenges primarily arose while extracting granular information for the BPMN models. Since the number of stakeholders involved in the elicitation activities was substantially high, incompleteness in the reports may be due to tacit knowledge issues, i.e., the LL coordinators did not include information that was assumed to be obvious, given their knowledge of the LL. Furthermore, LL coordinators complained about the fine-grained level of detail required by the reporting templates, which was, however, deemed necessary to achieve meaningful and complete diagrams. This is in line with previous research highlighting the difficulty in reaching clarity and understanding in elicitation interviews [11] with additional challenges in the agricultural domain [7, 12]. Other approaches focus on technology-intensive crowd RE techniques to enhance data collection, e.g., with mobile apps for farmers [12]. In our case, the gaps were filled by the domain expertise of the modellers—a goal that the software engineers involved admit they could not have achieved. Although this does not rule out further validation with stakeholders, the involvement of domain experts can speed up the process of reaching a consensus on the requirements.

💡 Domain expertise is crucial when modelling in specific technical fields, as someone with less relevant knowledge may struggle to fill the somewhat inevitable gaps in the elicited requirements.

L5 – End-users can resolve doubts by resorting to their peers The agronomists decided to work autonomously rather than in joint work sessions. However, whenever they faced challenges—whether in interpreting particular aspects of the reports or in utilising the modelling tools—they turned to peers who complemented their specific knowledge with additional viewpoints and expertise. This highlights that establishing a team of peers is particularly useful for modelling. As noticed by Abrahão [4], collaborative modelling tools could further enhance modelling speed and accuracy.

💡 Successful modelling benefits from teamwork, where different perspectives, skills, and insights are integrated to address gaps in knowledge and ensure a comprehensive understanding.

L6 – End-users prioritise visual effectiveness over formality. From the evaluation of the models, a general formal accuracy was observed, albeit the representations contained some recurrent issues (cf. Table 3, col. Revision). These included conceptual errors in the use of elements of the notations, element redundancy or cohesion, verbosity and use of non-standard names and styles. Notably, the models were delivered without end-users seeking clarification from the software engineers, or expressing doubts about the outcome. During the follow-up discussion, the agronomists confirmed that they were sometimes aware of the deviations and provided different motivations, e.g., personal preferences and use of symbols that were considered more aesthetically appropriate, although semantically incorrect. They also highlighted that they were first interested in expressing their understanding in the most visually effective way, and secondarily to adhere to the formal rules. The agronomists also remarked that relying on the final revision by software engineers for the technical soundness of the models enhanced their confidence in modelling.

💡 End-users are not afraid of making mistakes, prioritising effective visualisation over formal correctness. This suggests that iterations with software engineers are necessary to balance expressiveness with formality.

L7 – End-users demand easy-to-use and open-source modelling tools The agronomists reported that, when using the suggested tools and templates for drawing diagrams, they encountered obstacles at various levels. The process involved downloading and configuring various general-purpose tools, some of which were difficult to use, e.g., StarUML. Furthermore, there were limitations with the free versions and not all tools provided the comprehensive support needed to fully implement the procedure, e.g., for BPMN overlaps. In response to these challenges, an agronomist opted to use a licensed drawing tool that he was more familiar with—i.e., PowerPoint. The emerging difficulties are in line with previous research [4] highlighting the need to improve the user experience of modelling tools for diverse user groups. On the other hand, this highlights the adaptability of the proposed procedure to different tools, even those not specialised in modelling.

💡 The modelling procedure is sufficiently flexible to be implemented with different tools. However, modelling platforms should be open and adaptable to users with different skills. This would reduce technical barriers and better support end-users in focusing on the modelling task.

L8 – Requirements modelling helps agronomists to see the wide picture of digitalisation The agronomists agree that the modelling activity

helped them to understand the connections between digital technologies, social actors, and environmental resources. Models make clear the socio-technical nature of digital transformation and its impact in terms of novel activities/actors/resources to be introduced or removed from existing agricultural processes, due to the adoption of the technology. As a future step, the agronomists plan to use the representations produced to drive cost-benefit analyses, associating costs/benefits to the different components, and computing the sustainability of the solution.

💡 Models can be valuable tools for domain experts to explore, understand, and analyse digitalisation in ways that are directly relevant to their work, also enabling reflections on the impacts of technical solutions.

5 Conclusions

This paper reported an experience on end-user requirements modelling in the field of digital agriculture. Using a RE method based on MoDRE notations, a team of agronomists created iStar, UML, and BPMN models that describe the transformation occurring in four different contexts. In line with the design science approach [23], we considered both the perspective of the community of local practices, represented by the agronomists, and the research community, and we derived eight lessons that provide valuable insights into engaging end-users in formal modelling, an activity typically carried out by requirements engineers.

Our lessons emphasise the importance of having domain-specific knowledge to clearly understand agricultural processes and create accurate models. We also highlight that involving end-users in modelling may help to focus on more user-centred and sustainability-related aspects. Thus, the method could contribute to increase the social responsibility of developers by bringing into view the broader, often overlooked aspects, such as environmental and social dimensions. At the same time, we outline the challenges faced by the agronomists, such as the limitations of general-purpose tools, which underscore the need for more appropriate solutions for end-users, and the difficulty of the abstraction task. The content of the models also requires a final validation with stakeholders, including farmers and advisors, which may highlight additional issues not currently discovered. Further refinement may also be needed to support the foreseen cost-benefit analysis, and integrate standard practices in agricultural economics.

References

1. Codecs. <https://www.horizoncodecs.eu/>, accessed: 11-08-2024
2. Business process model and notation (BPMN) v2.0. <https://www.omg.org/spec/BPMN/2.0/> (2010), accessed: 11-08-2024
3. Unified modeling language (UML) 2.5.1 core specification. <https://www.omg.org/spec/UML> (2017), accessed: 11-08-2024
4. Abrahão, S., Bordeleau, F., Cheng, B., Kokaly, S., Paige, R., Störrle, H., Whittle, J.: User experience for model-driven engineering: Challenges and future directions (2017)

5. Assar, S.: Model driven requirements engineering: Mapping the field and beyond. In: 2014 IEEE MoDRE (2014)
6. Bano, M., Zowghi, D.: User involvement in software development and system success: a systematic literature review. In: EASE'13. pp. 125–130 (2013)
7. Doerr, J., Hess, A., Koch, M.: Re and society - a perspective on re in times of smart cities and smart rural areas. In: RE'18. IEEE (2018)
8. Felder, R.: Learning and teaching styles in engineering education. *Journal of Engineering Education -Washington-* **78**, 674–681 (01 1988)
9. Ferrari, A., Bacco, M., Gaber, K., Jedlitschka, A., Hess, S., Kaipainen, J., Koltsida, P., Toli, E., Brunori, G.: Drivers, Barriers and Impacts of Digitalisation in Rural Areas from the Viewpoint of Experts. *IST* **145**, 106816 (2022)
10. Ferrari, A., Lepore, F., Ortolani, L., Brunori, G.: Eliciting the double-edged impact of digitalisation: a case study in rural areas. In: RE '23. pp. 157–168. IEEE (2023)
11. Ferrari, A., Spoletini, P., Gnesi, S.: Ambiguity and tacit knowledge in requirements elicitation interviews. *REJ* (2016)
12. Groen, E.C., Rahman, K.R., Narsinghani, N., Doerr, J.: Towards crowd-based requirements engineering for digital farming (CrowdRE4DF). In: RE'24. pp. 457–465. IEEE (2024)
13. Guinee, J.: Handbook on life cycle assessment. an operational guide to the iso standards. *The International Journal of Life Cycle Assessment* (2001)
14. Horkoff, J., Aydemir, F.B., Cardoso, E., Li, T., Maté, A., Paja, E., Salnitri, M., Piras, L., Mylopoulos, J., Giorgini, P.: Goal-oriented requirements engineering: An extended systematic mapping study. *Requir. Eng.* **24**(2), 133–160 (jun 2019)
15. Leminen, S., Westerlund, M., Nyström, A.G.: Living labs as open-innovation networks. *Technology Innovation Management Review* **2**, 6–11 (09 2012)
16. Lieberman, H., Paternò, F., Klann, M., Wulf, V.: End-user development: An emerging paradigm. In: Lieberman, H., Paternò, F., Wulf, V. (eds.) *End User Development*, pp. 1–8. *Human-Computer Interaction Series*, Springer (2006)
17. Mannari, C., Bacco, M., Spagnolo, G.O., Malizia, A., Ferrari, A.: Towards a method for modelling socio-technical process transformation in digital agriculture. In: REW. pp. 306–315. IEEE (2024)
18. Mannari, C., Sportelli, M., Meesala, H., Okoye, O.F., Lepore, F., Bacco, M., Brunori, G., Malizia, A., Ferrari, A.: Documentation for paper "End-user Requirements Modelling: an Experience Report from Digital Agriculture" (2024). <https://doi.org/10.5281/zenodo.14168757>, accessed: 11-15-2024
19. Mannari, C., Sportelli, M., Okoye, O.F., Bacco, M., Ferrari, A., Malizia, A., Brunori, G.: Towards a toolkit for socio-technical process modelling in agriculture: a pilot study. In: *MetroAgriFor '24* (2024)
20. Mavin, A., Wilkinson, P., Teufl, S., Femmer, H., Eckhardt, J., Mund, J.: Does goal-oriented requirements engineering achieve its goal? In: RE'17. IEEE (2017)
21. Rijswijk, K., Klerkx, L., Bacco, M., Bartolini, F., Bulten, E., Debruyne, L., Dessen, J., Scotti, I., Brunori, G.: Digital Transformation of Agriculture and Rural Areas: a Socio-Cyber-Physical System Framework to Support Responsibilisation. *JRS* (2021)
22. Ryan, M., Isakhanyan, G., Tekinerdogan, B.: An interdisciplinary approach to artificial intelligence in agriculture. *NJAS* **25**, 1–31 (01 2023)
23. Wieringa, R.: *Design Science Methodology for Information Systems and Software Engineering* (2014)
24. Wing, J.: *Computational thinking*. *Communications of the ACM* (2006)
25. Yu, E., Giorgini, P., Maiden, N., Mylopoulos, J., Fickas, S.: *Social Modeling for Requirements Engineering*. The MIT Press (2011)